12th ESSLLI Student Session Proceedings
Preface

The ESSLLI Student Session has been part of the annual European Summer School in Logic, Language and Information ever since it was held for the first time in 1996. The Student Session is a unique interdisciplinary forum for student researchers from around the world to present their work in progress for a wide audience in a friendly environment.

This year’s Student Session in Dublin attracted a total of 51 submissions of which 16 were chosen to be presented as talks and 7 as posters. This volume contains all these papers.

The Student Session program committee consists mostly of students. Thus the Student Session provides a valuable practice ground not only for the students who present the results of their research but also for the ones who organize the event. The process of reviewing submissions requires seeking out and contacting senior researchers in various fields. This year’s program committee was able to collect three or more reviews for each submission, which provides excellent feedback to all the students who submitted their work.

We give our thanks to all the people of the previous Student Sessions who have provided us with helpful documentation, Ivana Kruijff-Korbayov, Carlos Areces, Amalia Todirascu, Raffaella Bernardi, Malvina Nissim, Kristina Striegnitz, Judit Gervain, especially to Sophia Katrenko and Janneke Huitink, the chairs of the previous Student Session, and to Carl Vogel for organizing the host event ESSLLI. A special mention must go to the program committee who played a crucial role in the whole organizing process.

Ville Nurmi and Dmitry Sustretov
Chairs of the ESSLLI 2007 Student Session

Helsinki and Nancy, June 2007
Program Committee

Chairs

Ville Nurmi, University of Helsinki
Dmitry Sustretov, INRIA Lorraine

Area experts

Paul Buitelaar, DFKI (Language and Computation)
Valentin Goranko, University of the Witwatersrand
(Logic and Computation)
Carl Pollard, Ohio State University (Logic and Language)

Co-chairs

Logic and Computation

Bryan Renne, City University of New York
Levan Uridia, University of Amsterdam

Language and Computation

Luciana Benotti, INRIA Lorraine
Michael Kaisser, University of Edinburgh

Logic and Language

Jana Häussler, University of Konstanz
Miltiadis Kokkonidis, University of Oxford
Program

First week
6–10 August

Monday, 6th
Suzan Verberne Evaluating Answer Extraction for Why-QA using RST-annotated Wikipedia texts
Ekaterina Lapshinova Extracting Predicates Subcategorizing for Wh-Clauses: an Architecture for a Semi-automatic System

Tuesday, 7th
Kevin Demiddele No future Adams pairs: applying the global/local conditional probability distinction
Yusuke Kubota, E. Allyn Smith A Multi-Modal Combinatory Categorial Grammar Analysis of Japanese Non constituent Clefting

Wednesday, 8th
Poster session

Thursday, 9th
Thomas Icard Towards An Alternative Proof of Solovay’s Arithmetical Completeness Theorem
Simone Bova A Bottom-Up Algorithm for t-Tautologies

Friday, 10th
Øistein E. Andersen Grammatical error detection using corpora and supervised learning
Voula Gotsoulia Foundations of Semantic Role Annotation: An Entailment-based Annotation Scheme
## Second week
13–17 August

### Monday, 13th
- **Bert Le Bruyn**
  - Partitivity in natural language
- **Eleni Kalyvianaki**
  - Factual Content in Algorithmic Natural Language Semantics

### Tuesday, 14th
- **Andrew Gargett**
  - An Incremental Model of Fragments in Dialogue
- **Olga Pustylnikov**
  - Guessing Text Type by Structure

### Wednesday, 15th
- **Camilo Thorne**
  - Managing Structured Data with Controlled English and Description Logics
- **Nicolas Troquard**
  - Some clarifications in logics of agency

### Thursday, 16th
- **Sabine Gründer**
  - Aspectual Shift via Supervaluation
- **Márta Peredy**
  - Obligatory adjuncts in weak accomplishments

### Friday, 17th
- **Closing event**
# Table of contents

Grammatical error detection using corpora and supervised learning  
Øistein E. Andersen  

Using Description Logics for Recognising Textual Entailment  
Paul Bedaride  

A Bottom-Up Algorithm for $t$-Tautologies  
Simone Bova  

Partitivity in natural language  
Bert Le Bruyn  

Statistical extension to the Poliqarp search engine  
Aleksander Buczyński  

No future Adams pairs: applying the global/local conditional probability distinction  
Kevin Demiddele  

Memory-Based Word Sense Disambiguation for Romanian with Automatic Feature Selection  
Georgiana Dinu  

An Incremental Model of Fragments in Dialogue  
Andrew Gargett  

Foundations of Semantic Role Annotation: An Entailment-based Annotation Scheme  
Voula Gotsoulia  

Aspectual Shift via Supervaluation  
Sabine Gründer
Towards An Alternative Proof of Solovay’s Arithmetical Completeness Theorem
Thomas Icard

Factual Content in Algorithmic Natural Language Semantics
Eleni Kalyvianaki

Applying a Focus Tree Model of Dialogue Context to Interactive Question Answering
Manuel Kirschner

Total Lexicalism in Language Technology
Judit Kleiber

A Multi-Modal Combinatory Categorial Grammar analysis of Japanese Nonconstituent Clefting
Yusuke Kubota, E. Allyn Smith

Extracting Predicates for Wh-Clauses
Ekaterina Lapshinova

Acquisition of Irregular Patterns in Spanish Verbal Morphology
Laia Mayol

An Approximate Gazetteer for GATE based on Levenshtein Distance
Bruno Woltzenlogel Paleo

Obligatory adjuncts in weak accomplishments
Márta Peredy

Guessing Text Type by Structure
Olga Pustylnikov

Managing Structured Data with Controlled English and Description Logics
Camilo Thorne
Some clarifications in logics of agency

Nicolas Troquard

Evaluating Answer Extraction for Why-QA using RST-annotated Wikipedia texts

Suzan Verberne
Grammatical error detection using corpora and supervised learning

OISTEIN E. ANDERSEN
University of Cambridge
oa223@cam.ac.uk

Abstract. This paper first describes how the Cambridge Learner Corpus can be enriched through parsing with RASP and used to train a binary sentence classifier. The results obtained are then analysed, which leads to the hypothesis that the sentence-level probabilistic approach might not be adequate. Experiments on simple error types verify that looking for specific errors within a sentence may indeed be a better approach. Further work includes building specialised classifiers for more complex errors and finding a way of combining the evidence from each.

1 Introduction

Traditionally, systems for grammatical error detection have relied upon hand-crafted rules. Atwell (1987) describes an early attempt to avoid this, and others have trained classifiers on artificial errors. Foster (2004) argues that genuine samples are needed, but the idea of training a classifier using real-life examples of incorrect constructions is merely suggested, as it would require a much larger corpus than the one Foster has compiled.

The Cambridge Learner Corpus provides a large quantity of correct and incorrect English text; moreover, the errors are identified and corrected, which makes a supervised learning approach feasible.

In this paper, we look at how evidence from corpora can be used efficiently and in combination with information acquired, e.g., through parsing, to find ungrammatical constructions, specifically limited to those that can be identified as such in absence of extra-sentential context.

2 Binary sentence classification

Reliably identifying each single error in a text, possibly suggesting corrections or indicating the type of error committed, is no simple task, even less so if the mistakes are numerous and interdependent. Verifying each word separately, on the other hand, would amount to little more than traditional
Grammatical error detection using corpora and supervised learning

spell checking. A reasonable compromise seems to be a binary classification scheme that distinguishes between correct and incorrect sentences, disregarding any correlation with the surrounding text.

This binary sentence classification problem can be approached with machine learning techniques, using features extracted from a set of sentences, each of which is defined as being either correct or incorrect, as training data for supervised learning. Formally, each sentence \( w \) is described by a set of features \( v_i \), constituting a feature vector \( v \), and associated with a class label \( \lambda \), indicating whether the sentence is correct or not. Characteristic features of a sentence may include single words, combinations of adjacent or grammatically related words and word classes, sentence length and parsing information.

For the sake of example, let us have a look at the underpinning of the naïve Bayesian classifier. According to Bayes’ rule, the probability that a sentence described by a feature vector \( v \) belong to a given class \( \lambda \), \( P(\lambda|v) \), can be reformulated thus:

\[
P(\lambda|v) = \frac{P(v|\lambda) P(\lambda)}{P(v)}
\]

Obviously, \( P(v) \) is constant for a given sentence and need not be evaluated if we are only interested in finding the most likely \( \lambda \). The conditional probability \( P(v|\lambda) \) is unfortunately difficult to estimate directly from training data, due to data sparseness (unique \( v \) for most distinct sentences unless the feature set is extremely restricted); the classical solution is to assume that all the features \( v_i \) in \( v \) are independent, an in this case relatively harmless,\(^1\) albeit obviously flawed assumption (the words constituting and hence the features describing a meaningful sentence are clearly not completely unrelated) that allows us to express the most likely \( \lambda \) as

\[
\hat{\lambda}_v = \operatorname{argmax}_\lambda P(v|\lambda) P(\lambda) = \operatorname{argmax}_\lambda P(\lambda) \prod_i P(v_i|\lambda).
\]

Training data was obtained from the *Cambridge Learner Corpus* (CLC, developed by Cambridge University Press), a continuously growing resource that currently contains over twenty million words of learner English written during language examinations. About half of the corpus has been manually inspected for errors, each of which has been annotated with the type of error committed and the words affected, as well as a suitable correction in the form of a replacement (unless the author’s intended meaning seemed unclear to the annotator).

\(^1\)Previous experiments have shown that the simple naïve Bayesian classifier outperforms several more complex classifiers that avoid this assumption, and that even the better ones give only marginally better performance (Andersen, 2006).
Table 1.1: These are the most important features used in the best-performing system in Andersen (2006). The examples are extracted from the sentence fragment \textit{went} \textit{VVD to} \textit{II the} \textit{AT cinema} \textit{NN}, where the subscripts are part-of-speech tags given by the RASP system, which also provides grammatical relations. Furthermore, two binary features indicate the event of a parsing error (no complete parse found) or the use of a fragment rule (the parser's last resort, used when no 'proper' grammatical rule applies), in which case the PoS tags of the words concerned are added as additional features. Different features combining the same atomic information, \textit{e.g.}, a word bigram and a grammatical relation between the same two words, are kept distinct.

Thanks to the annotation, both correct and incorrect versions of each sentence can be extracted. Parsing the sentences with RASP (Briscoe, Carroll and Watson, 2006) provides complementary information about the individual words (lemmatised forms, part of speech), uncovers the grammatical relations between them and gives a few general characteristics concerning the sentence as a whole. Judicious combinations of these different pieces of information as features for the classifier should provide it with the knowledge needed to separate right from wrong.

### 3 Results and comments

Different variations on the general approach outlined above are described and evaluated in Andersen (2006). The best system quoted, a pure naïve Bayesian classifier trained on 80\% of the data using the features summarised in Table 1.1 without smoothing or other enhancement techniques, gives an overall accuracy of 70\% on the test set; \textit{i.e.}, 70\% of the sentences, half of which are correct and half of which are incorrect, are correctly identified as such by the classifier.

The accuracy is highly dependent upon the type of error, as indicated in Fig. 1.1. Spelling mistakes obtain the highest detection rate, closely followed by inflectional and derivational errors, all of which tend to result in intrinsically malformed words. Verbal tense errors are located at the opposite end of the spectrum, and the system hardly does better on errors involving replacement, insertion or deletion of words; these errors are clearly harder
Figure 1.1: The hatched bars represent the total number of errors of each category in the test set, and the white bars represent the number of errors occurring in sentences identified as incorrect by the classifier. The percentage shows the proportion of errors found.

Figure 1.2: The hatched bars indicate the total number of sentences with a given number of errors in our test set, and the white bars indicate the ones that are correctly classified as erroneous. The percentage is the proportion of incorrect sentences correctly identified as such.

to spot, not only because the mistake consists in a prohibited combination of individually correct words, but also because the wrong sentence may be grammatically correct, merely conveying a message different from the one the writer intended.

A different break-down of the accuracy is shown in Fig. 1.2, which clearly indicates that a higher number of errors in a sentence makes it more likely to be caught by the system. This is hardly surprising, given that a sentence with multiple errors will contain many discernible features, whereas a sentence with one single, possibly minor, mistake will have much in common with a perfectly correct sentence. Conversely, a correct sentence does not differ substantially from a slightly incorrect one and thus may mistakenly be classified as wrong.

The strong effect on performance caused by the mere quantity of errors in a given sentence suggests that we may want to look at potential errors individually, which would also make it possible to use more directed approaches to various kinds of errors. An entire sentence would then be classified as
Øistein E. Andersen

Classification

<table>
<thead>
<tr>
<th></th>
<th>Correct</th>
<th>Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct ( p &gt; 0.5 )</td>
<td>162,394</td>
<td>166</td>
</tr>
<tr>
<td>Incorrect ( p &gt; 0.5 )</td>
<td>51</td>
<td>199</td>
</tr>
</tbody>
</table>

Table 1.2: Sentence-level classification. Results shown as number of sentences. The correct sentences include those that do not feature the indefinite article, but they are all classified as correct by the classifier, so this does not affect precision and recall of incorrect sentences.

incorrect if at least one error was identified with a high degree of confidence.

4 Towards a more refined approach — *a/an*

In the sentence-classification experiment described, there are many sources of noise and several things going on at the same time, which makes it difficult to identify the different contributions and draw clear conclusions. We therefore wanted to verify the rôle played by the number of errors per sentence in a more controlled setting.

For this experiment, we chose to look at the relatively simple problem concerning the form of the indefinite article, which shall be either *a* or *an* depending on whether the following sound is consonantal or vocalic. There is a certain discrepancy between spelling and pronunciation (consider, e.g., *an hour, an M.P.* and *a European*), so the correct form is not immediately obvious given the following letter. We therefore extracted two distinct features for each occurrence of the indefinite article, 1) the article combined with the following letter and 2) the article combined with the following word.

Thanks to the CLC, both correct and incorrect examples could be extracted and used for training.

The initial result on our test set is indicated in Table 1.2 and corresponds to 55% recall of incorrect sentences with 80% precision, which is hardly impressive on such a simple task. When we looked more closely at the data, however, it turned out that many of the overlooked errors occurred within sentences with multiple instances of the indefinite article; in fact, 71% of the incorrect sentences that contain one or more additional, correct, instances of the indefinite article are incorrectly classified as correct, as are 82% of the sentences with two or more such instances. A possible explanation could then be that the evidence for an error was simply drowning in evidence for the contrary, making the sentence ‘predominantly correct’ and thus most likely ‘quite right’ as far as the classifier was concerned.

We then ran a new experiment, in which not entire sentences, but single occurrences of the indefinite article were to be classified as correct or incorrect. As indicated in Table 1.3, 95% of the incorrect occurrences were found, and very few correct occurrences were mislabelled (90% precision).
This seems to indicate that individual examination of potential errors may indeed be helpful.

A closer inspection of the classifier output shows that the classifier correctly classifies the vast majority of the data with high confidence \((p \geq 0.9)\) for the chosen class label, as indicated in Table 1.4; it misses only two errors \((^{*}a\ HTV\ and\ ^{*}a\ MC)\) and apparently gives ten false positives with this high degree of confidence. However, four of the ‘false’ positives turn out to be real errors overlooked by the annotators, one is due to a transcription error \((can\ transcribed\ as\ c\ an)\), and the remaining are occurrences of the letter \(a\) rather than the indefinite article, a problem that could have been avoided using part-of-speech tagging. The quasi-totality of the misclassifications are thus done with a lesser degree of confidence and mainly concern somewhat irregular or difficult cases like underground, universal, US and historic; these can be checked manually, or the classifier can be improved, e.g., through more training data or more salient features.

5 Determiner-noun agreement — this/these

After having demonstrated that concentrating on single errors could indeed be beneficial, we wanted to attack a slightly more complex error type, requiring some knowledge of grammar as opposed to mere juxtaposition of words. Agreement between determiner and noun was chosen as a well-defined, purely grammatical, sentence-internal problem that does not rely on the meaning of the words involved or other less clear-cut concepts, and we then focused on the misuse of this when these is needed, by far the most common confusion of this kind in the CLC.

(1) a. *this/these friends
    b. *this/these old school friends
(2) a. *This is what good friends do.*

b. *I need this/these for a meeting with good friends of hers.*

Example 1 shows that we really need to identify the noun determined by *this/these* and examine its number in order to be able to choose between the singular and the plural determiner. Moreover, *this/these* does not always determine a noun at all, as illustrated by Example 2, and in this case no determiner-noun agreement should be attempted.

The RASP parser is able to identify the grammatical relation between *friends* and *these* in both phrases in Example 1 and correctly refrains from establishing a direct link between *this/these* and any of the nouns in Example 2. Because the RASP system is aware of agreement rules, no connection will be made between the plural noun *friends* and the incorrect singular determiner *this* in Example 1, but this is, alas, not a reliable indication of error, given that *this/these* may well appear in isolation. A possible solution is to make RASP ignore the requirement of number agreement between noun and determiner; then, the system will be allowed to link, e.g., *this* with *friends*, and the presence of a grammatical relation between a singular determiner and a plural noun (or vice versa) should be a good error indication.

We parsed the sentences containing *this* with this slight modification to the parser and found that the impossible relation between singular determiner and plural noun was established in 77% of the sentences containing an incorrect instance of *this* and in only a very low proportion of the full set of correct sentences (see Table 1.5). Quite a few of the incorrect instances were actually impossible to spot due to interactions with other errors in the text, e.g., *this job* corrected to *these jobs*; we therefore re-evaluated the system’s performance on the set of incorrect sentences in which the determiner agreement error appears in isolation, which gave a recall of 92% or 197 out of 215 incorrect uses of *this* correctly identified as such. The remaining 8% were missed partly due to parsing or tagging errors (e.g., an instance of the plural noun *treasures* was tagged as a third person singular verb form, which

<table>
<thead>
<tr>
<th>Relation involving <em>this</em></th>
<th>Correct</th>
<th>Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>No link to plural noun</td>
<td>48,590</td>
<td>326</td>
</tr>
<tr>
<td>Link to plural noun</td>
<td>326</td>
<td>1,105</td>
</tr>
</tbody>
</table>

Table 1.5: Presence or absence of a link between *this* and a plural noun in correct and incorrect instances of the singular determiner.

---

2 If, on the other hand, no grammatical relation contains *this*, then something is probably amiss. Only a small portion (at most 20% in our experiments) of the incorrect sentences can be found by exploiting this directly, though, for the parser will often find another (incorrect) grammatical relation involving *this*. 

7
in turn prevented the correct analysis from being found), and partly due to real number ambiguity (this/these people).

6 Further directions

These experiments can tell us something about how to build a better system for classification of sentences according to their grammaticality: The results obtained on a/an and this/these seem to consolidate the idea that specialised detection schemes for different kinds of errors may indeed yield better performance than one general classifier trained on all error types simultaneously. We believe that a sensible set of such expert detectors can be combined to form a system that will be able to detect incorrect sentences with high precision and good recall; as an added bonus, the crux of the problem can often be highlighted.

(3) a. *different/various courses
b. *have/incur a fine
c. aware *about/of something
d. depend *of/on something

The particular error types that have been hitherto discussed are quite obviously simpler than most, as they can be defined easily in terms of inviolable grammatical rules; subtler or more complex errors will clearly necessitate more sophisticated approaches to be detected in a reliable way. The expressions in Example 3 illustrate that the perceived incorrectness of an expression may well be due to lack of idiomaticity rather than evident ungrammaticality. These are cases where comparison with a large collection of correct English text like the British National Corpus (BNC) (Leech, 1992) may come to the rescue; more precisely, the choice of one word instead of another is in some cases governed by usage rather than inherent meaning (consider, e.g., the use of specific prepositions with particular verbs or the distribution of quasi-synonyms), and a large corpus should give a good indication of such strong collocations which should normally be respected.

Finally, it should be noted that some of the errors indicated in the CLC cannot realistically be expected to be found by a machine, even less so when each sentence is regarded in isolation. An example of this is the sentence I went to the cinema [...] recently, which has been corrected to I will go to the cinema [...] soon, presumably because the past tense is incompatible either with surrounding text or with the relevant exam question. However, this limitation hardly undermines the merits of a system for automatic detection of a wide variety of grammatical errors in the broadest sense.
Acknowledgements

We would like to thank Cambridge Assessment and Cambridge University Press for having granted us access to the Cambridge Learner Corpus as part of the English Profile Project (http://www.englishprofile.org/index.html). This paper reports on research supported by the University of Cambridge ESOL Examinations.

Bibliography


Cambridge University Press: Cambridge Learner Corpus ⟨http://www.cambridge.org/elt/corpus/learner_corpus2.htm⟩.


Grammatical error detection using corpora and supervised learning
Using Description Logics for Recognising Textual Entailment

PAUL BEDARIDE
Loria – UHP Nancy 2 – Talaris
paul.bedaride@loria.fr

Abstract. The aim of this paper is to show how we can handle the Recognising Textual Entailment (RTE) task by using Description Logics (DLs). To do this, we propose a representation of natural language semantics in DLs inspired by existing representations in first-order logic. But our most significant contribution is the definition of two novel inference tasks: A-Box saturation and subgraph detection which are crucial for our approach to RTE.

1 Introduction

Recognising textual entailment (RTE) is performing the following task: given two texts $T_1$ and $T_2$ in natural language, determine if we can infer $T_2$ from $T_1$. As an example consider the three following sentences:

A: “Adam has a son who has a son”,
B: “Adam has an offspring who has an offspring”,
C: “Adam is a grandfather”

We can infer B from A because a son is an offspring. We can also say that B and C are equivalent because a grandfather is a male who has an offspring who has an offspring and because Adam is a male name. But we cannot infer A from B or C because an offspring can be a son or a daughter.

As we can see, recognising textual entailments is far from trivial, involving many issues that are difficult to solve. The main issue is that natural language is highly expressive. Due to this expressivity, it is possible to express the same meaning in several ways, as in B and C. Furthermore modifying, adding or deleting a word in a sentence can completely change its meaning (e.g., Adam (dis)likes Eve). Another important issue, related to the first, is that there exists a huge number of synsets (i.e., sets of words with the same meaning). It is difficult to exactly map the relation among them (e.g., an offspring is a child) and to represent all background knowledge
needed for detecting textual entailment. However, as the RTE task is widely
considered to be relevant for such tasks as Question-Answering, information
retrieval, multi-document summarization and information extraction, the
task has received a great deal of attention in recent years.

Several different approaches to this task have been proposed and some
of them have been compared in the RTE Pascal challenge [1]. This chal-
lenge compares the different approaches using a corpus of annotated pairs
of texts, usually referred to as T for Text and H for Hypothesis. For each
pair, it is specified whether T entails H or not. One outcome of this com-
parison is that symbolic methods perform better than statistical methods.
Symbolic methods — using techniques and intuitions rooted in semantics,
syntax, logic, etc — typically have about 75% accuracy. Statistical meth-
ods — based on techniques like n-grams, lexicon, etc — have about 60%
accuracy. The method that we describe in this paper is symbolic. It dif-
fers from other symbolic methods because it uses Description Logics. The
first important reason to choose these logics for the RTE task is that they
are decidable and there exists highly optimized reasoners (e.g., RACER [2])
for different inference tasks. Moreover, we can (at least partially) represent
background knowledge and the semantic representation of sentences in these
logics. Other symbolic techniques which have already been investigated for
the RTE task (e.g., lexical alignment to detect synonyms), could perhaps
be integrated into our approach, improving the performance, but we are not
going to discuss these possibilities here.

As Description Logics (DLs) are the core of our approach to textual en-
tailment, we will start with a very brief introduction to these formalisms.
Description Logics are formal languages for knowledge representation. They
were inspired by Quillian’s semantic network [3] and Minsky’s frame seman-
tics [4]. DLs classify knowledge in two parts: the T-Box and the A-Box.
The T-Box contains terminological information which is general (good for
representing background knowledge). The A-Box contains assertions which
are specific (good for representing sentences). Another way to see the divi-
sion between these two kinds of information is to regard the T-Box as rules
which govern our world (e.g., laws from physics, chemistry, biology, etc),
and the A-Box as depicting the world’s individuals (e.g., a table, a chair, a
man, etc).

Description Logics employ the notions of concept, role and individual.
Concepts are classes of elements and are interpreted as a subset of a given
universe. Roles are links between elements and are interpreted as binary
relations of a given universe. Individuals are the elements of a given universe.

A knowledge base Σ is a pair \( (T, A) \). \( T \) is the T erminalogical-Box, a
finite set of expressions called General Concept Inclusions (CGI) with shape
\( C_1 \sqsubseteq C_2 \) where \( C_1, C_2 \) are concepts. The intended meaning of \( C_1 \sqsubseteq C_2 \)
is that the set of individuals in \( C_1 \) is included in the set of individuals in
$C_2$. $C_1 \sqsubseteq C_2$ is a notation for $C_1 \subseteq C_2$ and $C_2 \subseteq C_1$. Formulas of $T$ are also called terminological axioms. $A$ is the A(SSERTion)-Box, a finite set of expressions with shape $a:C$ or $(a, b): R$ where $C$ is a concept, $R$ a role and $a, b$ two individuals. The first expression means that the individual $a$ belongs to the set of individuals satisfying $C$. The second expression means that the relation $R$ holds between $a$ and $b$. Formulas of $A$ are called assertions.

In the description logic that we used, which is known as $\mathcal{ALC}T$ [5], we can form complex concepts from atomics concepts. They can be made up by negation ($\neg$), conjunction ($\sqcap$) and disjunction ($\sqcup$) of concepts. Roles can either be atomic, or the inverse ($R^{-}$) of an atomic role. We can also use the universal quantifier ($\forall \text{Role} \cdot \text{CONCEPT}$) to form a complex concept which is true for an individual $i$ if all roles Role which have for first argument $i$, have for second argument an individual for whom CONCEPT is true. The existential counterpart is defined as in first order logic: $(\exists \text{Role} \cdot \text{CONCEPT}) \equiv (\neg \forall \text{Role} \cdot \neg \text{CONCEPT})$.

Several reasoning tasks can be handled in DLs once we have defined a knowledge base $\langle T, A \rangle$. For example, instance checking tests if an individual is an instance of a specified concept. Relation checking tests if there exists a relation between two individuals. Knowledge base consistency tests if $\langle T, A \rangle$ is consistent. These tasks can be used for defining more complex tasks such as query individuals which find all instances of a concept.

We will define two novel reasoning tasks to use DLs for RTE. The most important of these is the subgraph detection task, which we will discuss in detail later; here we’ll introduce the simpler A-Box saturation task. This consists of completing A-Box information according to a given T-Box. Given a knowledge base $\langle T, A \rangle$, we say that $A'$ is a saturation if for each individual $a$, atomic concept $C$ and role $R$ appearing in $\langle T, A \rangle$ there is an assertion $a:C$ in $A'$ if and only if $\langle T, A \rangle \models a:C$, and an assertion $(a, b):R$ in $A'$ if and only if $\langle T, A \rangle \models (a, b):R$.

For example we can have the following T-Box:

$$T = \left\{ \begin{array}{l}
\text{PARENT} \triangleq \exists \text{Parent-of.SOMEONE} \\
\text{GRANDFATHER} \triangleq \exists \text{Father-of.PARENT}
\end{array} \right\}$$

expressing respectively that parent is equivalent to someone who is the parent of someone and grandfather is equivalent to someone who is the father of someone who is a parent. We can also represent the sentence “Adam is the father of someone who is the parent of someone” by the following assertions:

$$A = \left\{ \begin{array}{l}
a: \text{ADAM}, s_1: \text{SOMEONE}, s_2: \text{SOMEONE} \\
(a, s_1): \text{Father-of}, (s_1, s_2): \text{Parent-of}
\end{array} \right\}$$

By applying the T-Box to the A-Box we can deduce that $s_1$ is a PARENT thanks to the first rule and that $a$ is a GRANDFATHER thanks to the second rule. If we add the two pieces of information to the A-Box, we obtain a saturated A-Box.
There exist automatic theorem provers for different DLs including ALCI the logic we are going to use. They handle efficiently several reasoning tasks, including instance and relation checking, concept and knowledge base consistency, and getting all instances of a concept. They can also perform A-Box saturation, but the more complex subgraph detection task will require a new algorithm.

2 Representation of sentences in DLs

To start with, we will explain how the meaning of a sentence can be partially represented as a DL formula. We say partially because the expressivity of DLs is limited and the meaning of a text is complex, so our representation is an approximation of the actual meaning of the text. For instance, many syntactic elements such as articles, quantifiers, and modalities will not be considered in our approach. The sentence “The cat eats an apple”, for example, will be approximated by “cat eat apple”.

During the definition of our representation we should remember that our final goal is recognising textual entailment, hence we should struggle to have the same representation for the same meaning whenever possible. The main idea of our approach is to represent each sentence by an A-Box, the background knowledge by a T-Box and then to check if the model of the entailed sentence is a subgraph of the graph of the entailing sentence.

We now describe our approach step by step. We first discuss how to represent sentences in DLs. We start by introducing predicate-arguments dependencies, then we discuss modifiers, and we finish by explaining adjectives and negation.

Predicates-arguments dependencies. Our representation of sentences is based on Davidson’s semantics [6] which represents events as individuals. For example, the sentence “John loves Mary” is represented by the first order formula $\text{love}(e) \land \text{john}(j) \land \text{mary}(m) \land \text{agent}(e,j) \land \text{patient}(e,m)$. Here $e$ stands for the event of loving, $j$ stands for the individual named John, and $m$ stands for the individuals named Mary. $j$ is the agent of the event $e$, and $m$ is its patient.

By using Davidson’s semantics we only need to use unary or binary predicates. This fits well with our DLs approach by making a correspondence between unary predicates and concepts, and binary predicates and roles. The sentence “John loves Mary” is represented in DL as e:LOVE, j:JOHN, m:MARY, (e, j):Agent, (e, m):Patient.

We have agreed then on a semantic form, but we do not know which set of basic concepts and roles we will use for representing the meaning of words and the relations between words. To define our signature (i.e., the set of basic concepts and roles), we use the linguistic database FrameNet [7]
based on frame semantics. FrameNet is composed of semantic frames which involve frame elements and which are evoked by certain lexical units. For instance, the \textit{Commercial transaction} frame describes a common situation involving a buyer, a seller, some goods and some money and it is evoked by such words as buy, sell, pay, cost, spend, etc.

To specify the signature which allows us to represent verb semantics, we link the frame semantics to our representation, and we link frames to concepts which represent the sense of verbs, and frame elements to relations which connect verbs to their arguments.

For example, when we want to represent a verb like sell, we start by looking up in FrameNet the corresponding frame. FrameNet tells us that the concept sell is represented by the frame \texttt{COMMERCIAL TRANSACTION} and by the thematic relations Buyer, Seller, Goods and Money. Then for the sentence “Adam buys chocolate in the supermarket for 2 euros”, we have the following representation as a DL A-Box:

$$A = \left\{ \begin{array}{l}
\text{ct:COMMERCIAL TRANSACTION} \\
a:ADAM, s:SUPERMARKET, c:CHOCOLATE, p:2\_EUROS \\
\end{array} \right\}$$

\textbf{Verb modifiers.} The meaning of a verb can be affected by modifiers (e.g., place, time, manner, etc). For example in the sentence “The dog barks loudly”, loudly affects the meaning of bark by adding to it the fact that the sound produced by the bark is noisy. We must be able to say that “The dog barks loudly” entails that “The dog barks” but not the converse. Verbs may have many modifiers of the same type, but this is not a problem with Davidson-style representations. For each modifier we simply conjoin a concept which represents the modifier sense to the A-Box individual corresponding to the verb. For example, the sentence “John bought a car on Monday 8 may at 5pm” has the following representation:

$$A = \left\{ \begin{array}{l}
\text{ct:COMMERCIAL TRANSACTION} \cap MONDAY \cap 8\_MAY \cap 5PM \\
j:JOHN, c:CAR \\
(ct,j):Buyer, (ct,c):Goods
\end{array} \right\}$$

\textbf{Adjectives.} In Davidson’s approach, adjectives are represented as unary predicates applied to the variable which represents the word to which the adjective is applied. This representation can easily be used in DLs for adjectives that modify nouns; such adjectives are essentially treated in the same way that verb modifiers are. But adjectives can also occur following the copula as in “The cat is big”. How do we treat them? As our final goal is to recognize textual entailment, we have to be able to check that “The big cat” is equivalent to “The cat is big”.

The simplest way to recognise textual entailment is to have the same representation for the same meaning. We will thus represent adjectives and
the verb *to be* in the same way. That is, for adjectives we add a concept representing the adjective to the event individual which represents the word to which the adjective is applied. And we consider the verb *to be* as an isolated verb: we do not create any individual for it, but we add to the individual representing the subject of the verb the concept representing the verb’s copula. That is, “the big cat” and “the cat is big” will be represented in the same way, by: \(c: \text{CAT} \sqcap \text{BIG}\).

**Negation.** Even though, we have negation in our representation language, modeling natural language negation is difficult. The problem is scope. For example, for the sentence “The dog doesn’t bark loudly” there are two possible interpretations. In the first interpretation, negation takes narrow scope and applies to *loudly*. In this reading we mean that the dog barks but that it doesn’t bark loudly. In the second interpretation, negation takes wide scope and applies to *bark loudly*. It means that we do not know if the dog barks, but if it barks it doesn’t do it loudly.

Scope is an ubiquitous phenomenon in natural language. Besides negation, it also plays a role for quantifiers and verb arguments (e.g., “John sees the girl with the telescope”). We can try to analyse all possibilities, but this soon leads to an exponential blowup (e.g., two negations in a sentence can give rise to four different interpretations, three negation to eight different meanings, and so on). Moreover, we must have a YES or NO answer for the RTE task, hence what should we do if the possibilities do not all agree?

Our choice of representation for negation is motivated by our mechanism for recognising textual entailment. This mechanism is a mix between logical implication and syntactic similarity. Let’s analyse a concrete example. We take the following sentences: (A) “John didn’t buy a fruit”, (B) “John didn’t buy a fruit at midnight”, (C) “John didn’t buy an apple”, and (D) “John didn’t buy a big fruit” because they represent the most common kinds of scope negation. With a standard reading of the sentences we are able to detect the following entailments (and only those): (A) \(\Rightarrow\) (B), (A) \(\Rightarrow\) (C) and (A) \(\Rightarrow\) (D).

To detect (A) \(\Rightarrow\) (B), we must have a logical implication between the negation of the verb (i.e., “buy”) and the negation of the verb and its modifiers (i.e., “buy at midnight”). So we must have a scope for the negation on verb and its modifier, because otherwise we won’t detect \(\neg \text{BUY} \sqsubseteq \neg (\text{BUY} \land \text{MIDNIGHT})\). To detect (A) \(\Rightarrow\) (C), we must have a logical implication between the concept \(\text{FRUIT}\) and the concept \(\text{APPLE}\). Lexical knowledge will give us the implication \(\text{APPLE} \sqsubseteq \text{FRUIT}\), but we need the contraposed form \(\neg \text{FRUIT} \sqsubseteq \neg \text{APPLE}\). So we must have the negation of concepts associated with verb objects to detect this textual entailment. Finally, to detect (A) \(\Rightarrow\) (D), we must have a logical implication between the negation of the verb arguments (i.e., “fruit”) and the negation of the verb arguments and
their adjectives. This is similar to the first case, so we have a scope for the negation on the verb arguments and their adjectives \(\neg (\text{FRUIT} \land \text{BIG})\).

## 3 Representing knowledge

Now that we have seen how to represent sentences in DLs by encoding them into the A-Box, we will see how we use background knowledge to detect textual entailments such as “a cat eats” ⇒ “an animal eats”. The knowledge required to detect this entailment is lexical knowledge which explains that a cat is an animal, thus that the CAT concept is subsumed by the ANIMAL concept.

We use two repositories of lexical knowledge to detect textual entailment. The first is FrameNet, which we already used to represent text with the same meaning in the same way. The second is WordNet [8], which records different lexical relations between synsets, like synonymy, antonymy or hyponymy. To check that T entails H, we retrieve slT and slH, the synsets list of the words of T and H using WordNet. For each synset st and sh of slT and slH we check if there exists a lexical relation between them. If there exists a synonymy relation between st and sh we add the following CGI to the background knowledge: \(\text{ST} \equiv \text{SH}\). If there exists an antonymy relation between st and sh we add the following CGI to the background knowledge: \(\text{ST} \sqsubseteq \neg \text{SH}\) and \(\text{SH} \sqsubseteq \neg \text{ST}\). And finally for the hyponymy relation we have three different cases. If st is an hyponym of sh then we get the following CGI: \(\text{SH} \sqsubseteq \text{ST}\). If sh is an hyponym of st then we get \(\text{ST} \sqsubseteq \text{SH}\). And if st and sh share an hyponym we get \(\text{ST} \sqsubseteq \neg \text{SH}\) and \(\text{SH} \sqsubseteq \neg \text{ST}\).

For example, to detect the textual entailment “a cat eats” ⇒ “an animal eats” we check lexical relations between senses of cat and animal using WordNet. We get that cat is an hyponym of animal and we obtain the CGI: \(\text{CAT} \sqsubseteq \text{ANIMAL}\). By applying this CGI to the representation of “a cat eats” we obtain the following saturated A-Box for the sentence “a cat eats”:

\[
A = \{ i: \text{INGESTION}, c: \text{CAT} \sqcap \text{ANIMAL}, (i, c): \text{Ingestor} \}\]

## 4 Inference detection - Subgraph detection

We have now a way to represent sentences and use background knowledge to detect textual entailment, and this brings us to the second, and more complex, of our novel inference tasks: subgraph detection. It remains to specify how we check if a sentence entails another sentence. To understand what this involves, we must first note that a saturated A-Box can be represented as one or more oriented and labeled graphs (see, for example Figure 1.1). What we call subgraph detection is divided into three steps. First, we create A-Boxes for the pair (T,H). Then we saturate them with the T-Box created...
Using Description Logics for Recognising Textual Entailment

Figure 1.1: The graph $H$ is a subgraph of the graph $T$

by using WordNet. Finally we traverse the graphs corresponding to these saturated A-Boxes to check if the second is a subgraph of the first. By doing this we verify if all the information in $H$ is also in $T$. The algorithm is shown on Figure 1.2.

We need to do this because existing theorem provers for DLs focus on tasks which involve one A-Box and one T-Box. There is no existing tool which handles relations between two DL knowledge bases, and this is what we required for RTE.

To illustrate our algorithm, we use the example in Figure 1.1 which aims to show the entailment between the sentence $T$: “John buys a cat at the pet shop for 50 euros” and the sentence $H$: “A shop sells an animal for 50 euros to John”. These sentences are represented by the following A-Boxes:

$$T = \{ ct1:COMMERCIAL\_TRANSACTION \}
\{ j1:JOHN, ps1:PET\_SHOP, c1:CAT, p1:50\_EUROS \}

$$H = \{ ct2:COMMERCIAL\_TRANSACTION \}
\{ j2:JOHN, s2:SHOP, a2:ANIMAL, p2:50\_EUROS \}

We compute the background knowledge for detecting the entailment between these two sentences by using WordNet and we obtain the following T-Box:

$$BK = \{ CAT \sqsubseteq ANIMAL \}$$
$$\{ PET\_SHOP \sqsubseteq SHOP \}$$

By applying this background knowledge to the DLs representation of the sentences $T$ and $H$ we obtain the graphs of the Figure 1.1.

Now we need to check whether a graph $GH$ is a subgraph of another graph $GT$. Our approach is divided in two parts: node checking and arc checking. The first step consists in checking that there exists a function $f$ which links to each node $NH$ of $GH$ a node $NT$ of $GT$ such that the concept associated to $NH$ is subsumed by the concept associated to $NT$. The next
Paul Bedaride

```python
deftype BIJ = Dict: {IND -> IND}
deftype UNBIJ = Dict: {IND -> (List : IND)}

/* the main function */
def main(t: ABOX, h: ABOX): BOOL is
    bij: BIJ // nodes that have just one correspondence
    unbij: UNBIJ // nodes that have more than one correspondence
foreach ind in h.getIndividuals do // get bij and unbij
    ind = t.indSatisfying(ind.concepts)
    if len(ind)>1 then
        unbij[ind] = ind
    elif len(ind)==1 then
        bij[ind] = ind
    else
        print "no correspondence for individual " + ind.name
    stop
return testAllBijection(bij, unbij)

/* find all bijections and test them */
def testAllBijections(bij: BIJ, unbij: UNBIJ): BOOL is
    if len(unbij)<0 then
        return testBijection(bij) // we have a bijection and we test it
    else
        ind, List[Ind] = unbij.pop
        foreach i in List[Ind] do
            if testAllBijections(bij+(ind.[i]), unbij) then
                return True
        unbij.append(ind, List[Ind])
    return False

/* test if with this bijection the entailment is correct */
def testBijection(bij: BIJ): BOOL is
    foreach (src, trg, name) in h.getRelations do // test all relations
        if not t.hasRelation(bij[src], bij[trg], name) then
            return False
    return True
```

Figure 1.2: Algorithm for subgraph detection

step is to check for each arc $\Lambda$ of $\mathbf{G}_H$ between nodes $N_1$ and $N_2$ if there is an arc between $f(N_1)$ and $f(N_2)$ in $\mathbf{G}_T$ which have the same label as $\Lambda$.

Now that we know how to check if a graph $\mathbf{G}_H$ is a subgraph of a graph $\mathbf{G}_T$, we will check if the graph of the sentence $\mathbf{H}$ is a subgraph of the graph of the sentence $\mathbf{T}$. The first step is to find if there exists a function $f$. In our example, finding this function is easy, and it is defined like this: $f(\text{ct2}) = \text{ct1}$, $f(\text{j2}) = \text{j1}$, $f(\text{a2}) = \text{c1}$, $f(\text{s2}) = \text{ps1}$, $f(\text{p2}) = \text{p1}$. Now for the second step we must check arcs, and we can see easily that the arcs of the graph of the sentence $\mathbf{H}$ exist in the graph of the sentence $\mathbf{T}$ via the function $f$. For instance, the arc $\text{Buyer}$ between $\text{ct2}$ and $\text{j2}$ exists between $f(\text{ct2})$ and $f(\text{j2})$, that is to say $\text{ct1}$ and $\text{j1}$.

We have used a simple example, but subgraph checking works with more complex graphs. By more complex graphs we mean graphs containing identically labelled nodes, or more than one relation between two nodes. The limit of our algorithm is when we have existentially quantified information, because in the saturated A-Box we do not expand existentials. So if we compare the saturated A-Boxes of “Adam is the father of someone who is a parent of someone” and “Adam is a grandfather” we will have many nodes in the first sentence and only one in the second. Thus the first sentence implies the second but not the converse.
5 Tests and Conclusion

To test our algorithm we have made an implementation in Python which uses the DL prover RACER. The application takes a file as input which contains pairs T,H of texts which have been annotated by hand with respect to whether T entails H or not, and it generates the semantics by using the C&C Tools and Boxer [9], and adds the relevant lexical knowledge to detect entailment using WordNet. After computing all this information, it will use the algorithm we describe in the previous section to test if T entails H. We have tested our algorithm on PARC sentence pairs from the University of Illinois at Urbana-Champaign, which contains 76 pairs selected to show relevant issues important to the textual entailment. We can test our implementation on only 75 pairs, as the semantic generation step fails in one of them. We obtain the following results:

<table>
<thead>
<tr>
<th></th>
<th>By Hand</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>True</td>
<td></td>
</tr>
<tr>
<td>Application</td>
<td>True</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>False</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>41</td>
<td>34</td>
</tr>
<tr>
<td></td>
<td>False</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>75</td>
</tr>
</tbody>
</table>

These tests have shown that what we do works for what we want to do. That is to say, it works for detecting entailments between simple sentences (with verbs, noun, verb modifiers, noun modifiers and negation), with simple lexical knowledge. As we said at the start of the paper, the present system is not intended to handle entailments which need complex knowledge, or entailments which hold due to modality, time expressions, quantification or counting. The incorrect cases in the test set were usually of this kind.

Our approach is limited by the expressivity of our representation, which handles only a tiny fragment of the English language. Due to the expressivity of DLs, some fragments of English will be hard to represent. For instance, modality needs ideas from modal logic to be represented correctly (e.g., “John is an alleged murderer” is represented by the formula $john(j) \land alleged(murderer(j))$ in neo-Davison’s semantic). Nevertheless, by using more expressives description logics, we can handle some other fragments, such as articulate connective examples (e.g., “if Mary comes, then John comes too”).

Currently we are working on the implementation of a syntactic analyser which translates text into our DL representation by using FrameNet, and testing the use of more expressive logics. For instance, we can use the one-of operator $O$ [5] to have constraints on labelled nodes in terminological axioms. This could be useful for representing sentences with disjunction

---

1 At present the system doesn’t use FrameNet, and instead we take the verb as concept and basic roles as agent and patient. Because of this we will miss converse cases (i.e., to sell and to buy) in our test.
on individuals like “John loves Mary or Jane”. We can also use hybrid logics [10] as \( \mathcal{H}(\@) \) for having more expressive constraints on labelled nodes, and to represent articulate connectives.

**Bibliography**


Using Description Logics for Recognising Textual Entailment
A Bottom-Up Algorithm for \(t\)-Tautologies

SIMONE BOVA
Department of Mathematics and Computer Science
University of Siena, Pian dei Mantellini 44, 53100 Siena, Italy
bova@unisi.it

Abstract. Fuzzy logics based on residuated \(t\)-norms provide a robust mathematical formalism for logical deduction under uncertain or vague premises. In this paper, we describe a decision algorithm for the tautology problem of Basic Logic, which is the logic of continuous \(t\)-norms and their residua [Háj98, CEGT00]. Our algorithm is a refinement of the semantic method of Baaz, Hájek, Montagna, and Veith [BHMV02].

1 Introduction

Imagine designing a family of propositional logics that satisfies the following list of requirements:

(i) the propositional variables: \(p_1, p_2, \ldots\), are interpreted over the real unit interval \([0, 1]\), linearly ordered by \(\leq\) in the usual way (fuzzyness);

(ii) the logical symbols: \(\bot\) (falsum), \(\odot\) (fuzzy conjunction), and \(\rightarrow\) (fuzzy implication), are respectively interpreted over the constant 0 and the binary functions \(f_\odot\) and \(f_\rightarrow\) on \([0, 1]\) (truth functionality);

(iii) \(f_\odot\) is associative, commutative, monotone and continuous;

(iv) \(f_\odot(x, 1) = x\) and \(f_\rightarrow(x, y) = 1\) if and only if \(x \leq y\), so that the restrictions of \(f_\odot\) and \(f_\rightarrow\) to \(\{0, 1\}\) behave like Boolean conjunction and implication;

(v) the fuzzy modus ponens rule, \(A \odot (A \rightarrow B) \vdash B\), is sound.

In this scenario, the pairs of operations known as \(t\)-norms and residua provide suitable interpretations for fuzzy conjunction and implication. Indeed, a continuous \(t\)-norm \(*\) is a continuous binary function on \([0, 1]\) that is associative, commutative, monotone \((x \leq y\) implies \(x * z \leq y * z\)) and has 1 as unit \((x * 1 = x\)). Given a continuous \(t\)-norm \(*\), the associated
A Bottom-Up Algorithm for $t$-Tautologies

The residuum is the binary function on $[0, 1]$ uniquely determined by the condition $x \rightarrow^* y = \max\{z : x \ast z \leq y\}$. Notice that $x \leq y$ is equivalent to $x \rightarrow^* y = 1$, and implies $y \rightarrow^* z \leq x \rightarrow^* z$ and $z \rightarrow^* x \leq z \rightarrow^* y$. Notice also that the fuzzy modus ponens is sound, since by definition $x \ast (x \rightarrow^* y) \leq y$, and powerful, in the sense that the value of $y$ is lower bounded by the maximal value of $x \ast (x \rightarrow^* y)$ which preserves the requirement of soundness.

Hence, a $t$-norm $\ast$ naturally determines a propositional fuzzy logic $L^*$ satisfying requirements (i)-(v) above. Formally, let $[0, 1]_\ast = ([0, 1], \ast, \rightarrow^*, 0)$ be the algebra over $[0, 1]$ equipped with the $t$-norm $\ast$ and its residuum $\rightarrow^*$. We call $[0, 1]_\ast$ the $t$-algebra of $\ast$. Then, $L^*$ is the propositional logic on the connectives $\odot, \rightarrow$ and the constant $\perp$ respectively interpreted on $[0, 1]_\ast$ as $\ast, \rightarrow^*$ and 0 (over this basis, $\neg A$ and $\top$ are defineable via $A \rightarrow \perp$ and $\neg \perp$, respectively). The tautologies of $L^*$ are the formulas evaluating to $1$ on $[0, 1]_\ast$ under any valuation of the variables in $[0, 1]$.

Interestingly, the Hilbert calculus $BL$ (Hájek’s Basic Logic) given by the axioms:

1. $(A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C))$
2. $(A \odot B) \rightarrow A$
3. $(A \odot B) \rightarrow (B \odot A)$
4. $(A \odot (A \rightarrow B)) \rightarrow (B \odot (B \rightarrow A))$
5a. $((A \rightarrow (B \rightarrow C)) \rightarrow ((A \odot B) \rightarrow C))$
5b. $((A \odot B) \rightarrow C)) \rightarrow ((A \rightarrow (B \rightarrow C))$
6. $((A \rightarrow B) \rightarrow C) \rightarrow (((B \rightarrow A) \rightarrow C) \rightarrow C)$
7. $\perp \rightarrow A$

and the rule $A \odot (A \rightarrow B) \vdash B$, turns out to be the logic of all continuous $t$-norms and their residua. That is, $BL \vdash A$ if and only if, for all continuous $t$-norms $\ast$, $A$ is a tautology of $L^*$ [Háj98, CEGT00]. In this light, we formalize the $t$-tautology problem as follows:

**Problem:** $t$-TAUT = $\{\langle A \rangle : BL \vdash A \} \subseteq \{0, 1\}^*$

**Input:** $\langle A \rangle \in \{0, 1\}^*$

**Output:** 1 if and only if $\langle A \rangle \in t$-TAUT

where $\langle A \rangle \in \{0, 1\}^*$ is a binary encoding of $A$ of length polynomial in the complexity of $A$, size$(A)$, which is the number of connectives occurring in $A$. For technical reasons, we put size$(\top) = 0$.

As a stronger result, Agliano and Montagna [AM03] have shown that $A \in t$-TAUT if and only if $A$ is a tautology with respect to the interpretation of the propositional language into a special $t$-algebra, defined as follows.
Definition 1. The algebra $\omega[0,1]_L$ is the algebra on the support $[0, +\infty]$ equipped with the operations $\ast$, $\rightarrow^*$ and the constants $0, +\infty$, where $\ast$ and $\rightarrow^*$ are respectively defined by:

$$x \ast y = \begin{cases} \min\{x, y\} & \text{if } |x| \neq |y| \\ \max\{|x|, x + y - |x| - 1\} & \text{if } |x| = |y| < +\infty \\ +\infty & \text{if } x = y = +\infty \end{cases}$$

$$x \rightarrow^* y = \begin{cases} y & \text{if } |y| < |x| \\ |x| + 1 - x + y & \text{if } |x| = |y| \text{ and } y < x \\ +\infty & \text{if } x \leq y \end{cases}$$

where $|x|$ is the integer part of $x$ and $|+\infty| = +\infty$.

A valuation (of the propositional language into $\omega[0,1]_L$) is a function $v$ such that $v(\bot) = 0$, $v(p_i) \in [0, +\infty]$ for all $i \in \mathbb{N}$, $v(A \odot B) = v(A) \ast v(B)$ and $v(A \rightarrow B) = v(A) \rightarrow^* v(B)$.

Theorem 1 (Aglianò and Montagna, 2003). $A \in t$-TAUT if and only if $v(A) = v(\top)$ for every valuation $v$.

The interpretation $\omega[0,1]_L$ allows to show that the complement of $t$-TAUT is (complete for) $\mathcal{NP}$, and, as a consequence, that $t$-TAUT is decidable [BHMV02]. Hence, it is natural to investigate decision algorithms for $t$-TAUT. In this paper, we present an algorithm, called BOTTOM-UP-BL, which is a refinement of the semantic method of Baaz, Hájek, Montagna, and Veith (BHMV-BL, in the sequel).

2 A Bottom-Up Algorithm for $t$-Tautologies

The present section introduces BOTTOM-UP-BL. After presenting the basic idea, patterned after BHMV-BL (Subsection 2.1), we describe in detail how BOTTOM-UP-BL works (Subsection 2.2), and we provide an example (Subsection 2.3). The main result of the paper is that BOTTOM-UP-BL is correct for $t$-TAUT (Subsection 2.4). Not surprisingly, the worst case running time of the algorithm is $\exp(n^{O(1)})$, where $O(n)$ bounds above the size of the input. For background on algorithms, we refer the reader to [CLRS01].

2.1 Idea

Any valuation $v$ determines a total order $\leq_A$ over the subformulas of $A$ (plus $\top$), stipulating that $B_1 \leq_A B_2$ if and only if $v(B_1) \leq v(B_2)$, for $B_1, B_2$ subformulas of $A$. Such an order satisfies either $A \leq_A \top$ or $A =_A \top$. However, there exist total orders of the subformulas of $A$ (plus $\top$) not corresponding to any valuation $v$. We call the former orders consistent, and the latter
A Bottom-Up Algorithm for \( t \)-Tautologies

Figure 1.1: (a) depicts the partition (given a formula \( A \)) of the set of orders into consistent orders, \( C \), inconsistent orders, \( U \), and locally inconsistent orders, \( L \) \( \subseteq U \). By definition, \( C \cup U \neq \emptyset \). BHMV-BL searches \( C \cup U \), while Bottom-Up-BL searches \( (C \cup U) \setminus L \). (b) and (c) depict the search spaces of Bottom-Up-BL, given in input (distinct) formulas \( A_1 \) and \( A_2 \). The gray regions, \( M_1 \) and \( M_2 \), are the set of the orders where \( A_1 \preceq_A \top \) and \( A_2 \preceq_A \top \) respectively. In the first case the output is 0 (\( M_1 \cap C_1 \neq \emptyset \)), in the second case the output is 1 (\( M_2 \cap C_2 \neq \emptyset \)).

orders inconsistent, respectively, sets \( C \) and \( U \) in Figure 1.1(a). As an example, if \( B_1, B_2, B_1 \rightarrow B_2, B_1 \odot B_2 \) are subformulas of \( A \), any order where \( B_1 \rightarrow B_2 \preceq_A B_2 \) or \( B_1 \preceq_A B_1 \odot B_2 \) is inconsistent, observed that any valuation \( v \) satisfies both \( v(B_2) \leq v(B_1 \rightarrow B_2) \) and \( v(B_1 \odot B_2) \leq v(B_1) \). Now, the important consequence of Theorem 1 is that the semantic consistency of a given order is computable in polynomial time. Hence, since \( C \cup U \) is finite, the algorithm BHMV-BL can check exhaustively all the orders for semantic consistency; the output will be 1 if and only if all consistent orders satisfy \( A =_A \top \) (equivalently, all the orders satisfying \( A \preceq_A \top \) are inconsistent). Our simple observation is that BHMV-BL approach allows for the following refinement: if we construct the orders inductively on the complexity of the subformulas of \( A \), starting from all the orders of the variables of \( A \), we can immediately detect some inconsistencies (applying Fact 1, see below), and therefore we can avoid the computation of a certain number of inconsistent orders (the set \( L \subseteq U \) of locally inconsistent orders in Figure 1.1(a)), improving the effectiveness of the computation (we guess that \( L \) is large).

More precisely, let \( A \) be a formula and \( S \) be the set of the subformulas of \( A \). Any valuation \( v \) determines a partition of \( S \cup \{ \top \} \) into \( h = |H| \) blocks, where \( H = \{ |v(B)| : B \in S \cup \{ \top \} \} \). Let \( b_1 < \cdots < b_{h-1} < +\infty \) be the natural total order over \( H \), let \( I = \{ \bot_1, \ldots, \bot_h \} \) be a set of fresh constant symbols (idempotents, in the sequel) and put \( v(\bot_j) = b_j \) for all \( 1 \leq j < h \) and \( v(\bot_h) = +\infty \). Now, \( v \) determines a total order \( \preceq_A \) over \( S \cup \{ \top \} \cup I \), stipulating that, for every pair \( B_1, B_2 \) of formulas in \( S \cup \{ \top \} \cup I \), \( B_1 \preceq_A B_2 \) if and only if \( v(B_1) \leq v(B_2) \).

**Notation 1.** Let \( B_1, B_2 \in S \). In the sequel, \( B_1 =_A B_2 \) is for \( B_1 \preceq_A B_2 \) and \( B_2 \preceq_A B_1 \), and \( B_1 \preceq_A B_2 \) is for \( B_1 \preceq_A B_2 \) and \( B_1 \neq_A B_2 \). Also, if there exists \( j \leq h \) such that \( B_1 \preceq_A \bot_j \preceq_A B_2 \), we write \( B_1 \ll B_2 \); if there exists
Simone Bova

\[ j < h \text{ such that } \perp_j \leq_A B_1 <_A B_2 <_A \perp_{j+1}, \text{ we write } B_1 \prec B_2; \text{ if there exists } j < h \text{ such that } \perp_j \leq_A B_1 \leq_A B_2 <_A \perp_{j+1}, \text{ we write } B_1 \equiv B_2. \]

As before, we say that an order \( \leq_A \) over \( S \cup \{ \top \} \cup I \) is consistent if and only if it corresponds to a valuation. Some inconsistencies follow immediately from Definition 1.

**Fact 1.** Let \( A, S \) and \( I \) as above, and let \( \leq_A \) be any total order over \( S \cup \{ \top \} \cup I \). Then, \( \leq_A \) is consistent only if it satisfies all the following statements (\( B_1, B_2, B_1 \odot B_2, B_1 \rightarrow B_2, C_1, C_2, C_1 \odot C_2, C_1 \rightarrow C_2 \in S \)):

(i) If \( B_1 \ll B_2 \) or \( \perp_j =_A B_1 \leq_A B_2 \) \((j \leq h)\), then \( B_1 \odot B_2 =_A B_1 \).

(ii) If \( B_1 \ll B_2 \), then \( B_1 \odot B_2 <_A B_1 \).

(iii) If \( B_1 \leq_A C_1 \) and \( B_2 \leq_A C_2 \), then \( B_1 \odot B_2 \leq_A C_1 \odot C_2 \). If in addition \( \perp_j <_A C_1 \odot C_2 \) and \( B_1 \rightarrow <_A C_1 \) or \( B_2 <_A C_2 \), then \( B_1 \odot B_2 <_A C_1 \odot C_2 \).

(iv) If \( B_1 \leq_A B_2 \), then \( B_1 \rightarrow B_2 =_A \top \).

(v) If \( B_2 \ll B_1 \), then \( B_1 \rightarrow B_2 =_A B_2 \).

(vi) If \( B_2 <_A B_1 \), then \( B_2 <_A B_1 \rightarrow B_2 \).

(vii) If \( B_1 \leq_A C_1 \) and \( C_2 \leq_A B_2 \), then \( C_1 \rightarrow B_2 \leq_A B_1 \rightarrow B_2 \). If in addition \( B_1 <_A C_1 \) or \( C_2 <_A B_2 \), then \( C_1 \rightarrow C_2 <_A B_1 \rightarrow B_2 \).

We insist that the condition above is necessary, but not sufficient (in general the inclusion \( L \subseteq U \) in Figure 1.1(a) is strict). The idea beyond BOTTOM-UP-BL is to exploit systematically Fact 1 to reduce the search space, avoiding the computation of locally inconsistent orders (compare the description of the iteration step given in Subsection 2.2).

### 2.2 Algorithm

We describe in detail the algorithm BOTTOM-UP-BL, commenting on the pseudocode below. The input to the algorithm is a formula \( A \), where \( a_1, \ldots, a_k \) are the atoms (subformulas of complexity 0) of \( A \) and \( \text{size}(A) = n \). Notice that, if \( \text{size}(A) = n \), then the variables of \( A \) are at most \( n+1 \).

**BOTTOM-UP-BL\((\langle A \rangle)\)**

1. \textbf{for } \( h \leftarrow 2 \text{ to } k+1 \)
2. \( o_A \leftarrow \perp_1 \leq_A \cdots \leq_A \perp_h =_A \top \)
3. \( w_A(\perp_j)_{1 \leq j < h} = 0, w_A(\perp_h) = w_A(\top) = 1 \)
4. \( p_A \leftarrow \emptyset \)
5. \( \text{ORD}_h \leftarrow \{(o_A, w_A, p_A)\} \)
6. \textbf{for } \( i \leftarrow 0 \text{ to } n \)
7. \( S \leftarrow \{E : E \text{ subformula of } A, \text{size}(E) = i\} \)
A Bottom-Up Algorithm for \( \ell \)-Tautologies

\[
\text{ORD}_i \leftarrow \text{extensions of } \text{ORD}_{i-1} \text{ to } S \text{ not excluded by Fact 1}
\]
\[
\triangleright \text{ Let } \text{ORD}_n = \{(o_{A,1}, w_{A,1}, p_{A,1}), \ldots, (o_{A,h}, w_{A,h}, p_{A,h})\}.
\]
\[
\text{if } (\exists 1 \leq m \leq f) A <_{A,m} \top \text{ holds and } p_{A,m} \text{ is feasible}
\]
\[
\text{output } 0
\]
\[
\text{output } 1
\]

Let \( 2 \leq h \leq k + 1 \), and let \( S, I \) be as above. Let \( S_{-1} = \{\top\} \cup I \) and \( S_i = \{E : E \in S \cup \{\top\} \cup I, \text{size}(E) \leq i\} \subseteq S \), for \( i = 0, \ldots, n \). In the sequel, for all \(-1 \leq i \leq n\), \( \text{ORD}_i \) is a finite set of triples of the form \((o_A, w_A, p_A)\), where:

(i) \( o_A \) (order, in the sequel) is the union of a relation \( E_1 <_A E_2 \) satisfying irreflexivity, antisymmetry and transitivity over \( S_i \), and a relation \( E_1 =_A E_2 \) satisfying reflexivity, symmetry and transitivity over \( S_i \), with the (technical) exception to the symmetry of \( =_A \) that there are not \( E \notin S_i \setminus I \) and \( \bot_j \in I \) satisfying \( E =_A \bot_j \). Moreover, \( o_A \) satisfies the chain \( E_1 q_1 \cdots q_{|S_i|} = E_2 \), where \( S_i = \bigcup_{p=1}^{|S_i|} E_p \) and \( q_p \in \{<_A, =_A\} \) for all \( p = 1, \ldots, |S_i| - 1 \). In the sequel, given an order \( o_A, \leq_A \) is for \( <_A \) or \( =_A \) and, if \( B_1 \) and \( B_2 \) are subformulas of \( A \), \( \min_A \{B_1, B_2\} \) is \( B_1 \) if \( B_1 \leq_A B_2 \) and \( B_2 \) otherwise.

(ii) \( w_A \) is a linear function over \( S_i \).

(iii) \( p_A \) is a set of linear equality and inequality constraints with integer coefficients over unknowns (among) \( x_1, \ldots, x_n, x_{n+1} \).

In lines 2-5 the algorithm settles \( \text{ORD}_h \) \((2 \leq h \leq k + 1)\) to the triple \((o_A, w_A, p_A)\), where \( o_A = \{\bot_1 <_A \bot_2, \ldots, \bot_{h-1} <_A \bot_h \} \cup \{\bot_h =_A \top\} \), \( w_A(\bot_j) = 0 \), for \( 1 \leq j < h \), \( w_A(\bot_h) = w_A(\top) = 1 \) and \( p_A = \emptyset \).

Now, let \( \text{ORD}_h \) be fixed. The main loop of Bottom-Up-BL spans lines 6-8. The \( i \)th iteration of the loop \((0 \leq i \leq n)\) is aimed to extend the triples in \( \text{ORD}_{i-1} \) to all subformulas of \( A \) of complexity less than or equal to \( i \) (stipulate that \( \text{ORD}_{-1} \) is \( \text{ORD}_h \)). In the description below, we consider several possible ways of extending each triple in \( \text{ORD}_{i-1} \), and we assume that the algorithm put every extension considered in \( \text{ORD}_i \).

Initialization (Step \( i = 0 \)). For each \((o_A, w_A, p_A) \in \text{ORD}_h \), the order \( o_A \) is extended to the atoms \( a_1, \ldots, a_k \) of \( A \) in such a way that:

(i) If \( \bot \) is an atom of \( A \), then \( \bot =_A \bot \) holds in the extended order.

(ii) For each \( 1 \leq j < h \), there exists an atom \( a \) of \( A \) such that \( \bot_j \leq_A a <_A \bot_{j+1} \) holds in the extended order.

(iii) There do not exist variables \( p \) of \( A \) such that \( p <_A \bot_j \) or \( \bot_h <_A p \) holds in the extended order. Notice that \( \bot_j =_A \bot \) can hold in the extended order, but \( p =_A \bot_j \) can not hold because of the previous stipulation on \( =_A \), for \( 1 \leq j \leq h \). There are several possible ways of extending \( o_A \) to the atoms. For each of such choices, \( w_A \) and \( p_A \) are
extended as follows. As regards to \( w_A \): if \( a \) is \( \bot \), then \( w_A(a) = 0 \), otherwise, if \( a \) is a variable \( p_i \), then \( w_A(a) = x_i \) \((i \geq 1)\). As regards to \( p_A \): (i) For each pair \( p_i, p_j \) such that \( p_i =_A p_j \), the constraint \( x_i = x_j \) is added to \( p_A \). (ii) For each pair \( p_i, p_j \) such that \( p_i <_A p_j \), the constraint \( x_i < x_j \) is added to \( p_A \). (iii) For each \( p_i \) such that \( \bot_j =_A p_i \), the constraint \( x_i = 0 \) is added to \( p_A \) if \( j < h \), otherwise the constraint \( x_i = 1 \) is added to \( p_A \). (iv) For each \( p_i \) such that \( \bot_j <_A p_i \), the constraint \( 0 < x_i \) is added to \( p_A \). (v) For each \( p_i \) such that \( p_i <_A \bot_j \), the constraint \( x_i < 1 \) is added to \( p_A \).

**Iteration (Step \( i + 1, \ i \geq 0 \).** Let \( S_i \), \( S_{i+1} \), \( ORD_i \) and \( ORD_{i+1} \) be determined as above. At iteration \( i + 1 \), the algorithm computes \( ORD_{i+1} \), given \( ORD_i \). Each triple in \( ORD_{i+1} \) is the result of the extension of a triple in \( ORD_i \) to all the subformulas of \( A \) of complexity \( i + 1 \). There are several possibilities to extend an order \( o_A \), over \( S_i \) to an order \( o_{A,i+1} \) over \( S_{i+1} \). Among all, the algorithm computes only the extensions considered below (\( w_{A,i} \) and \( p_{A,i} \) are extended accordingly). For a fixed \( (o_A, w_A, p_A) \in ORD_i \) and a fixed subformula \( E \) of \( A \) of complexity \( i + 1 \), **BOTTOM-UP-BL** works as follows. If \( E \) has the form \( B_1 \circ B_2 \), then:

\((\circ_1)\) If \( B_1 \ll B_2 \) or \( \bot_j =_A B_1 \leq_A B_2 \) for some \( j \leq h \), then \( o_A \) is extended to \( B_1 \circ B_2 \) by adding \( B_1 \circ B_2 =_A B_1 \). Also, \( w_A(B_1 \circ B_2) = w_A(B_1) \) is settled, and no constraint is added to \( p_A \).

\((\circ_2)\) If \( B_2 \ll B_1 \) or \( \bot_j =_A B_1 \leq_A B_2 \) for some \( j \leq h \), then \( o_A \) is extended to \( B_1 \circ B_2 \) by adding \( B_1 \circ B_2 =_A B_2 \). Also, \( w_A(B_1 \circ B_2) = w_A(B_2) \) is settled, and no constraint is added to \( p_A \).

\((\circ_3)\) Otherwise, let \( j < h \) be maximal such that \( \bot_j <_A B_1 <_A \bot_{j+1} \) and \( \bot_j <_A B_2 <_A \bot_{j+1} \). Then, \( o_A \) is extended to \( B_1 \circ B_2 \) in such a way that: (i) \( \bot_j <_A B_1 \circ B_2 <_A \min_A \{B_1, B_2\} <_A \bot_{j+1} \) holds in the extended order. (ii) For any pair \( C_1, C_2 \) of formulas, if \( B_1 \leq_A C_1 \), \( B_2 \leq_A C_2 \) and \( C_1 \circ C_2 \) has already been added in \( o_A \), then \( B_1 \circ B_2 \leq_A C_1 \circ C_2 \) holds in the extended order. Moreover, if \( \bot_j <_A C_1 \circ C_2 \) \((j < h)\), \( B_1 \leq_A C_1 \), \( B_2 \leq_A C_2 \) and at least one of the last two inequalities is strict, then \( B_1 \circ B_2 <_A C_1 \circ C_2 \) holds in the extended order. (iii) For any pair \( C_1, C_2 \) of formulas, if \( C_1 \leq_A B_1 \), \( C_2 \leq_A B_2 \) and \( C_1 \circ C_2 \) has already been added in \( o_A \), then \( C_1 \circ C_2 \leq_A B_1 \circ B_2 \) holds in the extended order. Moreover, if \( \bot_j <_A C_1 \circ C_2 \) \((j < h)\), \( C_1 \leq_A B_1 , C_2 \leq_A B_2 \) and at least one of the last two inequalities is strict, then \( C_1 \circ C_2 <_A B_1 \circ B_2 \) holds in the extended order. There are several possible ways of extending \( o_A \) to \( B_1 \circ B_2 \) satisfying the conditions above. For each choice, \( w_A \) and \( p_A \) are extended accordingly, as follows.

As regards to \( w_A \): if \( \bot_j =_A B_1 \circ B_2 \) for some \( j < h \), then \( w_A(B_1 \circ B_2) = 0 \) is settled, otherwise \( w_A(B_1 \circ B_2) = w_A(B_1) + w_A(B_2) - 1 \) is settled.

As regards to \( p_A \): (i) If \( \bot_j =_A B_1 \circ B_2 \) for some \( j < h \), then the
A Bottom-Up Algorithm for ℰ-Tautologies

classification constraint $w_A(B_1) + w_A(B_2) \leq 1$ is added to $p_A$. (ii) If $\bot_j <_A B_1 \circ B_2$ for some $j < h$, then the constraint $1 < w_A(B_1) + w_A(B_2)$ is added to $p_A$. Also, if $B_1 \circ B_2 =_A D$ for some formula $D$ already added to $o_A$, the constraint $w_A(B_1 \circ B_2) = w_A(D)$ is added to $p_A$; otherwise, if the formulas $D_1, D_2$, already added to $o_A$, are respectively maximal such that $D_1 <_A B_1 \circ B_2$ and minimal such that $B_1 \circ B_2 <_A D_2$, the constraints $w_A(D_1) < w_A(B_1 \circ B_2)$ and $w_A(B_1 \circ B_2) < w_A(D_2)$ are added to $p_A$.

If $E$ has the form $B_1 \rightarrow B_2$, then:

$(-1)$ If $B_1 \leq_A B_2$, then $o_A$ is extended to $B_1 \rightarrow B_2$ by adding $\bot_h =_A \top =_A B_1 \rightarrow B_2$. Also, $w_A(B_1 \rightarrow B_2) = 1$ is settled, and no constraint is added to $p_A$.

$(-2)$ If $B_2 \ll B_1$, then $o_A$ is extended to $B_1 \rightarrow B_2$ by adding $B_1 \rightarrow B_2 =_A B_2$. Also, $w_A(B_1 \rightarrow B_2) = w_A(B_2)$ is settled, and no constraint is added to $p_A$.

$(-3)$ If $B_2 < B_1$, then let $j < h$ be maximal such that $\bot_j \leq_A B_2$. Then, $o_A$ is extended to $B_1 \rightarrow B_2$ in such a way that: (i) $\bot_j \leq_A B_2 <_A B_1 \rightarrow B_2 <_A \bot_{j+1}$ holds in the extended order. (ii) For any pair $C_1, C_2$ of formulas, if $B_1 \leq_A C_1, C_2 \leq_A B_2$ and $C_1 \rightarrow C_2$ has already been added in $o_A$, then $C_1 \rightarrow C_2 \leq_A B_1 \rightarrow B_2$ holds in the extended order. Moreover, if at least one of the above two inequalities is strict, then $C_1 \rightarrow C_2 <_A B_1 \rightarrow B_2$ holds in the extended order. (iii) For any pair $C_1, C_2$ of formulas, if $C_1 \leq_A B_1, B_2 \leq_A C_2$ and $C_1 \rightarrow C_2$ has already been added in $o_A$, then $C_1 \rightarrow C_2 \leq_A C_1 \rightarrow C_2$ holds in the extended order. Moreover, if at least one of the above two inequalities is strict, then $B_1 \rightarrow B_2 <_A C_1 \rightarrow C_2$ holds in the extended order. Again, there are several possible ways of extending $o_A$ to $B_1 \circ B_2$ satisfying the conditions above. For each choice, $w_A$ and $p_A$ are extended accordingly, as follows. As regards to $w_A$, $w_A(B_1 \rightarrow B_2) = w_A(B_2) + 1 - w_A(B_1)$ is settled. As regards to $p_A$: (i) If $B_1 \rightarrow B_2 =_A D$ for some formula $D$ already added to $o_A$, the constraint $w_A(B_1 \rightarrow B_2) = w_A(D)$ is added to $p_A$. (ii) Otherwise, let the formulas $D_1, D_2$, already added to $o_A$, be respectively maximal such that $D_1 <_A B_1 \rightarrow B_2$ and minimal such that $B_1 \rightarrow B_2 <_A D_2$. Then, if $D_1 <_A \bot_{j+1}$, the constraint $w_A(D_1) < w_A(B_1 \circ B_2)$ and $w_A(B_1 \circ B_2) < w_A(D_2)$ are added to $p_A$; otherwise, if $\bot_{j+1} =_A D_2$, only the constraint $w_A(D_1) < w_A(B_1 \circ B_2)$ is added to $p_A$.

Termination (Step $i = n$). The number of orders of the form $B_1 \leq_A \cdots \leq_A B_{(k+n)+1+h}$, where each $B_i$ is a distinct formula in $S \cup \{\top\} \cup I$ is clearly finite. The main loop of the algorithm computes a (proper) subset
of these orders for every fixed \( h \), so it terminates for every formula \( A \). Let \( \text{ORD}_n \) be the set computed at termination of the main loop, for some \( 2 \leq h \leq k + 1 \). For each triple \((o_A, w_A, p_A) \in \text{ORD}_n\), the order \( o_A \) contains \( \top \) and all the subformulas of \( A \), including \( A \) itself. If there exists a triple \((o_A, w_A, p_A) \in \text{ORD}_n\) such that \( A <_A \top \) holds in \( o_A \) and \( p_A \) is feasible (line 10), then Bottom-Up-BL breaks the external loop and outputs 0 (line 11). Otherwise, Bottom-Up-BL iterates the external loop if \( h \leq k \) (line 1), or outputs 1 if \( h > k \) (line 12).

### 2.3 Example

Let \( A \) be \(((p_1 \rightarrow \bot) \rightarrow \bot) \rightarrow p_1\). The atoms of \( A \) are \( p_1 \) and \( \bot \) and the subformulas of \( A \) excluding atoms, ordered by increasing complexity, are \( p_1 \rightarrow \bot \), \( (p_1 \rightarrow \bot) \rightarrow \bot \) and \( A \). For \( h = 2 \), we have \( \text{ORD}_h = \{(o_A, w_A, p_A)\} \) where \( o_A = \bot <_{A,1} \bot =_A \top \). At step \( i = 0 \), we have \( \text{ORD}_0 = \{(o_{A,1}, w_{A,1}, p_{A,1}), (o_{A,2}, w_{A,2}, p_{A,2})\} \) where \( o_A = \bot =_{A,1} \bot =_{A,1} \bot <_{A,1} 1 \leq_2 =_{A,2} \top =_{A,2} p_1 \). For \( h = 3 \), we have \( \text{ORD}_h = \{(o_A, w_A, p_A)\} \) where \( o_A = \bot <_{A,1} \bot <_{A,1} \bot <_{A,1} =_{A,1} \top \). At step \( i = 0 \), we have \( \text{ORD}_0 = \{(o_{A,3}, w_{A,3}, p_{A,3}), (o_{A,4}, w_{A,4}, p_{A,4})\} \) where \( o_{A,3} = \bot =_{A,3} \bot <_{A,3} \bot <_{A,3} \bot <_{A,3} \bot =_{A,3} \top \), and \( o_{A,4} = \bot =_{A,4} \bot <_{A,4} \bot <_{A,4} \bot =_{A,4} \bot =_{A,4} \top \). Now, for \( h = 3 \) and \( i = 1, 2, 3 \), Bottom-Up-BL computes, among the other possibilities, the following extension of \((o_{A,4}, w_{A,4}, p_{A,4})\) above, where \( w_{A,4}(\bot) = w_{A,4}(\bot) = w_{A,4}(\bot) = 0 \), \( w_{A,4}(p_1) = x_1 \), \( w_{A,4}(\bot) = w_{A,4}(\top) = 1 \), and \( p_{A,4} = \{0 < x_1, x_1 < 1\} \). By \((\rightarrow_2)\), subformula \((p_1 \rightarrow \bot) \rightarrow \bot \) adds \( p_1 \rightarrow \bot =_{A,4} \bot =_{A,4} \bot =_{A,4} \bot \) and settles \( w_{A,4}(p_1 \rightarrow \bot) = w_{A,4}(\bot) = 0 \) \((p_{A,4} \text{ is unchanged})\). By \((\rightarrow_1)\), subformula \((p_1 \rightarrow \bot) \rightarrow \bot \) adds \( \top =_{A,4} \bot =_{A,4} \bot =_{A,4} \bot =_{A,4} \bot \) and settles \( w_{A,4}(p_1 \rightarrow \bot) = w_{A,4}(\bot) = 1 \) \((p_{A,4} \text{ is unchanged})\). By \((\rightarrow_2)\), \( A \) adds \( A =_{A,4} p_1 =_{A,4} \top \) and settles \( w_{A,4}(A) = w_{A,4}(p_1) = x_1 \) \((p_{A,4} \text{ is unchanged})\). Hence, at termination, \( A =_{A,4} \top \) holds in \( o_{A,4} \) and \( p_{A,4} \) is feasible (any real number in \((0,1)\) is a solution to \( p_{A,4} \)), and Bottom-Up-BL outputs 0.

### 2.4 Correctness

Bottom-Up-BL is sound and complete for t-TAUT. Formally,

**Theorem 2.** Let \( A \) be a formula. Then, \( \langle A \rangle \in \text{t-TAUT} \) if and only if Bottom-Up-BL outputs 1 on input \( \langle A \rangle \).

The proof stems from the following correspondence between classes of valuations and triples \((o_A, w_A, p_A)\) with feasible \( p_A \)’s computed by Bottom-Up-BL. On the one hand, let \((o_A, w_A, p_A)\) be a triple computed by Bottom-Up-BL, where \( p_A \) is feasible. Let \( b = b_1 < \cdots < b_j < \cdots < b_{h-1} \) be any linear order of \( h - 1 \) nonnegative integers and let \( x = (x_1, \ldots, x_n, x_{n+1}) \) be any solution to \( p_A \) (there are several possible choices). Then, the valuation
A Bottom-Up Algorithm for $t$-Tautologies

$v$ corresponding to $(o_A, w_A, p_A)$ under $b$ and $x$ is such that, for $1 \leq i \leq n + 1$: if $\bot_h = A p_i$, then $v(p_i) = +\infty$; otherwise, if $j < h$ is maximal such that $\bot_j \leq_A p_i < A \bot_{j+1}$, then $\lfloor v(p_i) \rfloor = b_j$ and $v(p_i) - \lfloor v(p_i) \rfloor = x_i$. For definiteness, put $v(p_i) = 0$ for all $i > n + 1$. On the other hand, let $v$ be a valuation and let $v_0, \ldots, v_n$ be the restrictions of $v$ to the subformulas of $A$ of complexity $\leq 0, \ldots, \leq n$ respectively. Also, let $H = \{v(a_i) : 1 \leq i \leq k\}$, $h = |H| + 1$ and $b_1 < \cdots < b_j < \cdots < b_{h-1}$ be the natural total order of $H$. Now, extend $v$ to $\bot_1, \ldots, \bot_h, \top$ by putting $v(\bot_h) = v(\top) = +\infty$ and $v(\bot_j) = b_j$ for $1 \leq j \leq h - 1$. Then, for $i = 0, \ldots, n$ the triple $(o_{A,i}, w_{A,i}, p_{A,i})$ corresponding to $v_i$ can be computed mimicking iterations from 0 to $i$ of Bottom-Up-BL main loop with $h$ settled as above (the case $i = n$ gives the triple corresponding to the valuation $v$): $o_{A,i}$ is settled to the order $\leq_A$, determined by valuation $v_i$; $w_{A,i}$ and $p_{A,i}$ are settled in such a way that clauses $(\bigcirc_1)$, $(\bigcirc_2)$, $(\bigcirc_3)$ and $(-_1)$, $(-_2)$, $(-_3)$ are satisfied, with respect to the order $o_{A,i}$. Such a correspondence owns the following key property.

**Fact 2.** If a valuation $v$ corresponds to a triple $(o_A, w_A, p_A)$ computed by Bottom-Up-BL such that $A <_A \top$ holds in $o_A$ and $p_A$ is feasible, then $v(A) < v(\top)$. Conversely, if a triple $(o_A, w_A, p_A)$ computed by Bottom-Up-BL corresponds to a valuation $v$ such that $v(A) < v(\top)$, then $A <_A \top$ holds in $o_A$ and $p_A$ is feasible.

3 Conclusion

In this paper, we refined the decision algorithm for $t$-tautologies of Baaz, Hájek, Montagna, and Veith [BHMV02]. Specifically, we exploited an inductive construction to avoid the brute force computation of all the orders of the subformulas of the input formula. We mention two natural developments of the present work.

From the complexity point of view, it would be interesting to investigate the existence of a class $\mathcal{F}$ of formulas such that the set of locally inconsistent orders of any $A \in \mathcal{F}$ is provably large. Indeed, any $A \in \mathcal{F}$ would be easy for Bottom-Up-BL, but still hard for BHMV-BL. From the algorithmic point of view, it would be interesting to formalize a top-down refinement of BHMV-BL, patterned after the logical calculus presented in [BM07], and to compare its performances against those of the bottom-up refinement presented in this paper. In particular, [BM07] implies that the bound $\exp(n^{O(1)})$ can be improved to $\exp(3n/2)$, where $O(n)$ bounds above the size of the input.

**Acknowledgments.** The author thanks his supervisor Franco Montagna for stimulating discussions on the subject, and the anonymous referees for careful comments on the first version of this paper.
Bibliography


A Bottom-Up Algorithm for t-Tautologies
Partitivity in natural language

BERT LE BRUYN
Utrecht Institute of Linguistics OTS
Bert.LeBruyn@let.uu.nl

ABSTRACT. In this paper I will look at two analyses of partitives that incorporate the anti-uniqueness constraint in the semantics: [2] and [21], [22]. I will show that they present both conceptual and empirical problems and I will present a novel, pragmatic alternative. The main focus is on Zamparelli’s and my own analysis. Special attention is paid to a particular kind of partitive, viz. the faded partitive ([10]).

1 Introduction

In this paper I will look at two analyses of partitives that incorporate the anti-uniqueness constraint in the semantics: [2] and [21], [22]. I will show that they present both conceptual and empirical problems and I will defend a pragmatic alternative. The main focus is on Zamparelli’s and my own analysis. Special attention is paid to a particular kind of partitive, viz. the faded partitive ([10]).

2 Anti-uniqueness

[2] discusses the following contrast from Jackendoff:

1. *I met the [two of the men].
2. I met the [[two of the men] that you pointed out last night].

1Pseudo-partitives (e.g. A cup of tea) are not treated in this paper.
2I adapted the original example and I left out the parallel case of double genitives.
Partitivity in natural language

The fact that partitive constructions as in (1) cannot combine with the definite article is called the anti-uniqueness constraint on partitives. The way Barker accounts for the contrast is to assume that partitive of is not a realization of the improper part operator ($\leq$) but of the stronger proper part operator ($<$). How does this work?

If partitives are a realization of the proper part relation two of the men can only be defined if there are at least three men. Let’s assume that there are exactly three and call them Marc, Matthew and Luke. In these settings two of the men refers to $\{\{\text{Marc,Matthew}\},\{\text{Matthew, Luke}\},\{\text{Marc, Luke}\}\}$. Under the assumption that definites pick out the set with the highest cardinality if there is one and are undefined otherwise this explains the infelicity of (1). Indeed, in the case of two of the men there is no set with the highest cardinality and the definite will be undefined. The (possible) felicity of (2) follows straightforwardly. Let’s assume that only two men were pointed out last night: Marc and Luke. In these settings that you pointed out last night refers to $\{\{\text{Marc}\},\{\text{Luke}\},\{\text{Marc, Luke}\}\}$. The two of the men that you pointed out last night now refers to the set with the highest cardinality in the intersection of $\{\{\text{Marc, Matthew}\},\{\text{Matthew, Luke}\},\{\text{Marc, Luke}\}\}$ and $\{\{\text{Marc}\},\{\text{Luke}\},\{\text{Marc, Luke}\}\}$, viz. $\{\text{Marc, Luke}\}$.

It appears then that the assumption that partitive of is a realization of the proper part operator offers a very simple and elegant account of the contrast in (1) and (2). As pointed out by Barker it is moreover compatible with previous analyses of partitives in making the same empirical predictions. One could argue that there is a conceptual problem however. Whereas the improper part operator can be seen as the inverse of the join-operation a similar “natural” function does not seem to underlie the proper part operator. This is Zamparelli’s criticism. His implementation will be reviewed in the two following sections.

3 Zamparelli’s analysis of full partitives (Zamparelli 1998)

In this section I will evaluate Zamparelli’s analysis of full partitives (one of the boys, two of these girls) which contains a straightforward implementation of the anti-uniqueness constraint without using the proper part operator. In order to evaluate it I first have to define pluralities and definite determiners and lay out the syntactic structure Zamparelli assumes.

3.1 Plurality

3. $\langle -s \rangle$ = The power set of the set corresponding to the noun to which it is applied, minus the empty set.

In a model with four boys the denotation of boys is as follows:
4. $\llbracket \text{boys} \rrbracket = \{ \{a, b, c, d\}, \{a, b, c\}, \{b, c, d\}, \{a, c, d\}, \{a, b, d\}, \{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}, \{a\}, \{b\}, \{c\}, \{d\}\}$

3.2 Definites

5. Definite determiners denote the operator $\text{Max}$ which, when applied to a set, returns the element with the highest cardinality, if there is one, and it is undefined otherwise.

In the same model the result of applying $\text{Max}$ to the denotation of $\text{boys}$ is the following:

6. $\llbracket \text{the boys} \rrbracket = \text{Max}(\llbracket \text{boys} \rrbracket) = \{a, b, c, d\}$

Demonstratives differ from the definite article in introducing an extra restriction. Note furthermore that in a lattice approach $\text{Max}$ selects the supremum. For ease of exposition I will sometimes use this term.

3.3 Syntax of partitives

The syntactic structure Zamparelli assumes for partitives is the following (I got rid of projections that will not play a role):

7. $[DP \text{ two } [NP \text{ boys}, [RP \text{ of } [DP \text{ the } [NP \text{ boys}]]]]]$

Two properties stand out. The first is the fact that the downstairs (i.e. following of) NP has been copied to the upstairs (i.e. preceding of) NP position (this can hardly be called an analysis-specific assumption; most syntacticians working on partitives assume this (see [8], [12], [1], [5] and most recently [18]). The second is the special projection RP. This projection contains the ‘residue’ operator realized as of and is the semantic core of partitivity in Zamparelli’s analysis to which I turn presently.

Zamparelli’s analysis of partitives takes of to be the residue operator ($\text{Re}'$) which is defined as follows:

8. $\text{Re}'(A, b) = A - \{b\}$

Given the syntactic structure Zamparelli assumes it is not difficult to see how this operator gives us proper partitivity. It suffices to replace $A$ by the denotation of $\text{boys}$ and $\{b\}$ by the denotation of the boys. The result is the following:

9. $\llbracket \text{boys of the boys} \rrbracket = \{ \{a, b, c, d\}, \{a, b, c\}, \{b, c, d\}, \{a, c, d\}, \{a, b, d\}, \{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}, \{a\}, \{b\}, \{c\}, \{d\}\}$
Partitivity in natural language

This analysis derives proper partitivity and in order to do so it makes no use of the (conceptually) unattractive proper part operator. According to Zamparelli Re' can moreover be seen as a natural operator in the sense that it selects the complement set of a natural determiner, viz. the. Note though that the naturalness of this operator only comes about in Zamparelli’s application of Re’. There is nothing inherent to Re’ that makes it more natural than <.

The main problem I have with this analysis is one of compositionality. If one assumes that the upstairs copy is a copy of the downstairs noun the analysis makes wrong predictions. Take e.g. those boys and assume that in our model those boys refers to \{a, b\}. Applying Re’ blindly the result would be the following:

\[ [\text{boys of those boys}] = \{\{a, b, c, d\}, \{a, b, c\}, \{a, c, d\}, \{a, b, d\}, \{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\}, \{a\}, \{b\}, \{c\}, \{d\}\} \]

This e.g. wrongly predicts that four of those boys refers to \{a, b, c, d\} even though both c and d are not part of the denotation of those boys. In order for the analysis to give the right predictions Zamparelli has to assume (and he does) that what is copied to the upstairs position is not the NP with its modifiers but the contextually restricted set of boys of which the downstairs DP is the supremum. This means that one has to calculate what this set would be, restrict the set of boys accordingly and then apply the copy operation. This calls for a non-compositional analysis which, if possible, one would like to avoid.

4 Zamparelli’s analysis of faded partitives (Zamparelli 2002)

In this section I will evaluate Zamparelli’s analysis of faded partitives which picks up most of his analysis of full partitives. Before doing so I will however introduce the concept ‘faded partitives’ itself and define a few more notions I will be needing.

Faded partitives are sequences of the form of + DET + NOUN that can appear as such in argument position.\(^3\) At first sight they only seem to differ from full partitives in having no upstairs determiner. For reasons that are not important here the sequence DET + NOUN in faded partitives can however only to kinds and subkinds (see [4]). The kind interpretation is the one that is associated with the DP of faded partitives of the form of + the + NOUN and the subkind interpretation is the one that is associated with the DP of faded partitives of the form of + demonstrative + NOUN.

\(^3\)The term ‘Faded partitives’ was used already by [20].
The following expressions have been analyzed as faded partitives: *vandieN* ('of those N', Dutch) ([10], [16], [15], [4]), *de ces N* ('of those N', French) ([23]), *desN* ('of-the N', French) ([17]), *di questi N* ('of these N', Italian) ([13]), *dei N* ('of-the N', Italian) ([6], [19], [22], [3]). Here follow some typical examples from Italian:

11. Non accetto di questi commenti.
   Not I-accept of those comments
   I don’t accept this kind of comments.

12. Ho comprato dei biscotti.
   I-have bought of-the cookies
   I bought cookies.

Given their resemblance with full partitives it is generally assumed that faded partitives should be analyzed in the same way. This is what Zamparelli assumes too. Note though that he only treats *dei N* and not *di questi N*.

To present his analysis I will have to define the concepts 'kind' and 'subkind'. To do this I will use Chierchia’s down-operator that is defined as follows:

13. $\cap P$: (For any situation/world $s$) $\lambda s [\cap P_s]$ if $\lambda s [P_s]$ is in $K$, undefined otherwise ($P_s$ is the extension of $P$ in $s$)

Its inverse, the up-operator, is defined as follows:

14. $\cup d$: (Let $d$ be a kind. For any situation/world $s$) $\lambda x [x \leq d_s]$ if $d_s$ is defined, where $d_s$ is the plural individual that comprises all of the atomic members of the kind.

Kinds then receive the following definition:

15. The kind corresponding to a set $P$ is $\cap P$ (Zamparelli’s notation is $\{\langle \text{the kind}\rangle P\}$

Subkinds are nothing more than a kind to which an extra semantic restriction has been added$^4$:

16. $\cap \lambda x [P(x) & \text{Dem}(x)]$

I will now present Zamparelli’s analysis of faded partitives and afterwards point out the problems I find.

As could be expected Zamparelli wants to extend his analysis of full partitives. The gist of his analysis for full partitives is that

---

$^4$This is just one kind of subkind. To account for the subkind reading of indefinites one has to assume (see [7]) that next to the standard domain there exists a domain of subkinds. This kind of domain can however not be assumed to be underlying all subkind readings of demonstratives. The main problem this kind of analysis would have is to account for the fact that *those lions* can refer to one subkind of lions. (see [4])
the denotation of the downstairs DP gets subtracted from the denotation of the upstairs NP. Crucial here is that the upstairs NP denotes a set of which the downstairs DP is the supremum. The problem Zamparelli faces in faded partitives is that a kind-referring DP is the supremum of the singleton set that contains the kind itself. Maintaining the gist of his analysis of full partitives Zamparelli accordingly assumes the Re’ operation for the faded partitive of the boys looks as follows:

17. \{\text{(the kind) boys}\} - \{\text{(the kind) boys}\} = \emptyset

It should be clear that the result of the Re’ operation is not what one would want. Zamparelli is aware of this and proposes that as a last resort operation the up-operator can be applied to the first term yielding the following result:

18. \{x \mid \text{boy}(x)\} - \{\text{(the kind) boy}\} = \{x \mid \text{boy}(x)\}

Modulo some constraint on number and existential quantification \{x \mid \text{boy}(x)\} gives us the interpretation we want for of the boys, viz. ‘boys’ (see (12)).

Even though Zamparelli gets the facts right his analysis is problematic. Leaving aside many more problems I think the main thing that should be pointed out is that the way Zamparelli implements copying here is different from the way he implemented it in full partitives. The crucial difference is that in full partitives he could still defend that he copied the set corresponding to the downstairs noun. Here the copy is unmistakably the downstairs DP. I admit that in the case of full partitives he restricted the set in such a way that the downstairs DP was its supremum but the copy was still the set. The reason why he makes this move in faded partitives is that the kind boys is not the supremum of the set boys but only the supremum of the singleton set it is contained in. In order to keep the gist of his Re’ proposal, i.e. that the Re’ operation gets rid of the supremum, this move seems unavoidable. The move itself is however unacceptable; there is support for an NP copy but no support at all for a DP copy. One could explore two ways out. The first is to assume that what is copied is the downstairs noun. This would give the same result as in (18) but it would not be an implementation of an operation which gets rid of the supremum of a set. It would moreover be impossible to apply the same trick to di questi N because the upstairs noun cannot include the semantic restriction introduced by the demonstrative. The second way out goes as follows:

**Step 1**

Kind interpretation of the downstairs DP: \(\cap BOYS\)

**Step 2**

De-intensionalizing the downstairs DP: \(\iota BOYS = \{a, b, c, d\}\)

\[5\] I treat the competition with the bare plural elsewhere ([3]).
Step 3
Determining the set of BOYS the downstairs DP is the supremum of:
\{\{a, b, c, d\}, \{a, b, c\}, \{a, c, d\}, \{a, b, d\}, \{a, b\}, \{a, c\}, \{a, d\},
\{b, c\}, \{b, d\}, \{c, d\}, \{a\}, \{b\}, \{c\}, \{d\}\}.

Step 4
Copying the set determined in Step 3.

Step 5
Calculating the Residue operation:
\[\{\{a, b, c, d\}, \{a, b, c\}, \{a, c, d\}, \{a, b, d\}, \{a, b\}, \{a, c\}, \{a, d\},
\{b, c\}, \{b, d\}, \{c, d\}, \{a\}, \{b\}, \{c\}, \{d\}\} - \{a, b, c, d\}\]
\[\{\{a, b, c\}, \{a, c, d\}, \{a, b, d\}, \{a, b\}, \{a, c\}, \{a, d\}, \{b, c\},
\{b, d\}, \{c, d\}, \{a\}, \{b\}, \{c\}, \{d\}\}\]
Partitivity in natural language

analysis predicts though and, more crucially, what any analysis relying on anti-uniqueness would predict.

Given the empirical and conceptual problems the analyses incorporating anti-uniqueness in the semantics face I would like to look at an alternative in which we put anti-uniqueness in the pragmatics. This is worked out in the final section of this paper.

6 A novel way of looking at partitivity

In this section I will present a third way of looking at partitivity and anti-uniqueness. The gist is that we keep the (improper) part operator and explain the facts in (1) and (2) in the pragmatics. This allows us to avoid postulating the proper part operator and to use the standard up-operator for the analysis of faded partitives.

The way I take pragmatics to account for the facts in (1) and (2) is through application of a pragmatic principle like the following:

20. Avoid complexity:
   All other things being equal less complex expressions are preferred over more complex expressions.

This principle predicts that if we can find a shorter way of saying the two of the men we can explain why it is pragmatically odd to use it.\(^6\) It is not difficult to find such an expression, viz. the two men. Note that the two of the men you pointed out last night is not semantically equivalent to the two men you pointed out last night (at least not with the bracketing in (2)). What we then furthermore predict is that the two of the men is not uninterpretable semantically and that it might actually occur in language. The following attested examples show exactly this:

21. The two of the required course readings are: ...

22. Under Nash’s theory, either of the two of the equilibrium points is an equally ‘rational’ outcome.

23. To take this topic further a joint working group has been set up between the two of the International Energy Agency’s programmes - PVPS and the Solar Heating and Cooling Programme.

24. Some aspects of the START programme are also being used on courses in the two of the University’s Schools - School of Design Engineering and Computing and Institute of Health and Community Studies - in these cases the work undertaken will attract credit towards a University award.

\(^6\)The same approach is adopted by [11] who link the Avoid Complexity principle to Grice’s maxim of Manner.
Bert Le Bruyn

A question that still needs answering is where the proper partitivity effect comes from. I assume it originates in the use of the upstairs quantifier, e.g. *two in two of the men*. This assumption is based on the fact that quantifiers have the same effect outside full partitives as long as they quantify over a contextually restricted set. (25) e.g. is preferably interpreted as saying that only two out of a larger set of students were absent.

25. Two students didn’t show up in class.

Having proposed how we can account for the infelicity of *the two of the men* and where the proper partitivity effect comes from I can sketch the semantics I propose for full and bare partitives. As for full partitives, nothing prevents us from adopting an analysis as developed by [14] or [9]. As for faded partitives, I propose the following analysis in [3]:

26. of-those
\[ \forall z [\text{Dem}(z) \& \text{Lions}(z)] \]
\[ \lambda y \lambda x [\neg \exists y (y(x))] \]
\[ \lambda x [\forall \exists z (\text{Dem}(z) \& \text{Lions}(z))(x)] \]
\[ \text{of those lions} \]

27. of-the
\[ \forall z [\text{Books}(z)] \]
\[ \lambda y \lambda x [\neg \exists y (y(x))] \]
\[ \lambda x [\forall \exists z (\text{Books}(z))(x)] \]
\[ \text{of the books} \]

This analysis implements the improper partitivity relation for faded partitives. Both full and faded partitives then receive the same improper partitivity analysis.

7 Conclusion

In this paper I developed two arguments against analyzing full partitives as involving the proper part relation:

- Given that the improper part relation is not an option for faded partitives we would miss a generalization if we analyzed full partitives as involving the proper part relation.\(^8\)

- There is a straightforward pragmatic principle that accounts for the anti-uniqueness effects. The fact that there are attested examples where anti-uniqueness is not obeyed is an argument in favour of putting anti-uniqueness in the pragmatics.

The following counter-arguments were raised by the anonymous reviewers:

\(^7\)The details of the syntax are treated more explicitly in [4].

\(^8\)The same type of argument has been developed by [11] for pronominal and (vague) measure partitives.
Partitivity in natural language

- Given that faded partitives are semantically and syntactically clearly different from full partitives it is debatable whether we would miss a generalization.

- The attested examples in which anti-uniqueness is not obeyed are still odd and it is not clear in how far these examples are systematic.

I cannot but agree with the second counter-argument; the native English speakers I contacted agreed on the oddity of the examples. I however do not agree with the first. The reason for the disagreement is that the counter-argument seems to be based on the wrong assumption that faded partitives don’t allow for an upstairs quantifier (put differently: that full partitives don’t allow for kind-referring downstairs DPs). The fact that they do is easily missed if one only studies the of-the variants because quantifier + of-the + N is semantically equivalent to quantifier + N. Given this equivalence the avoid complexity principle rules out the more complex quantifier + of-the + N. The same does not apply to of-those variants because they are not equivalent to anything else.9 An example is given in (28):

28. Ik heb vier van die ventjes gezien.
    I have four of those little-guys seen

The partitive in (28) on the kind reading of the downstairs DP differs from more standard full partitives in two respects: (i) the downstairs DP refers to a kind and (ii) there is no proper partitivity effect. If I’m correct in my assumption that proper partitivity in full partitives originates in the use of a quantifier quantifying over a contextually restricted set (ii) follows from (i) (kinds are typically not contextually restricted). The kind reading and the non-kind reading of (28) can then be seen as a minimal pair showing that proper partitivity in full partitives is a pragmatic effect that originates in the use of a quantifier quantifying over a contextually restricted set. To come back to the first counter-argument raised by the anonymous reviewers: I hope to have shown that faded partitives and full partitives are not so different from one another that they would need a separate analysis.

Acknowledgements

I’d like to thank Henriëtte de Swart, Min Que and four anonymous ESSLLI reviewers for their comments on earlier versions of this paper. For the LATEX layout I’m deeply indebted to Christina Unger. For always being there for English judgments I thank Anna Asbury.

9Note furthermore that full partitives with kind-referring DPs are acceptable in more languages than those having faded partitives. This widens the scope of my argumentation.
Bibliography


Partitivity in natural language


Statistical extension to the Poliqarp search engine

ALEKSANDER BUCZYŃSKI

Institute of Computer Science
Polish Academy of Sciences
Ordona 21, 01-237 Warszawa, Poland
olekb@ipipan.waw.pl

ABSTRACT. Poliqarp is a utility for searching large tagged corpora with support for positional tagsets and ambiguous interpretations. The paper presents a statistical extension to its query language, which allows to ask for frequency distributions of specified attributes' values rather than contexts of the occurrences of each match. The extension also provides several statistical measures for collocation detection.

1 Introduction

1.1 Motivation

The aim of this paper is to present a recently implemented statistical extension to the Poliqarp corpus search engine [Janus and Przepiórkowski, 2006] developed at the Institute of Computer Science PAS.

Originally, Poliqarp was designed as a concordancer, responding to every query with a list of matches with contexts of selected width. This provides the user with examples of usage of specific constructions, but one can imagine many corpora problems in the case of which browsing through hundreds of occurrences is neither convenient nor efficient. The statistical extension introduces the possibility to easily find answers to questions like:

- what is the frequency distribution of a given word’s forms?
- what parts of speech may occur after a given word?
- what verbs are used most often in a given style?

The extension also provides several statistical measures for collocation detection, and for investigating correlations between individual attributes.
1.2 Poliqarp

Poliqarp is an open source utility for searching large tagged corpora, with an expressive query syntax and a fast search engine. A comparison of Poliqarp and its query language with other corpus search tools can be found in [Przepiórkowski et al., 2004], the most important novel features are:

- support for structured, externally defined tagsets, allowing easy access to individual morphosyntactic categories,
- support for ambiguous morphosyntactic interpretations, with distinction between certain and uncertain information.

The basic source format of the corpus assumed by Poliqarp is XCES — the XML Corpus Encoding Standard [Ide et al., 2000]. The search engine is currently employed in Polish [Przepiórkowski, 2004] and Portuguese [Barreto et al., 2006] corpora projects. The tagset may be specified externally and the internal character coding is UTF-8, so Poliqarp could be used for any corpus of any language. A stable version 1.0 is available to the community under the GNU GPL licence.

1.3 Terminology

In the remainder of this paper we assume the following terminology:

**segment** is the smallest interpreted unit, i.e., a sequence of characters with their morphosyntactic interpretations (lemma, grammatical class, grammatical categories);

**attribute** is a property of a single segment, like orthographic form, length of the orthographic form, base form, pos (grammatical class), a grammatical category (for example case or gender);\(^1\)

**pattern** is the first part of the query (before group by, but including within and meta); this is equivalent to Poliqarp query specified in [Przepiórkowski, 2004];

**match** is a sequence of segments matching a pattern;

**grouping rules** refer to the last part of the query (after group by).

\(^1\)Some attributes — orth, length, base, pos — are assumed to be universal for all corpora and segments (of course, value sets of the attributes may vary), others — like number, case, degree or gender — may depend on the tagset used in the corpus. The examples of queries given in this document are based on the tagset used in the IPI PAN Corpus of Polish.
Aleksander Buczyński

<table>
<thead>
<tr>
<th>orth number case</th>
<th>c(orth number case)</th>
</tr>
</thead>
<tbody>
<tr>
<td>woda sg nom</td>
<td>11285</td>
</tr>
<tr>
<td>wodach pl loc</td>
<td>848</td>
</tr>
<tr>
<td>wodami pl inst</td>
<td>682</td>
</tr>
<tr>
<td>wodą sg inst</td>
<td>5083</td>
</tr>
<tr>
<td>wodę sg acc</td>
<td>8709</td>
</tr>
<tr>
<td>wodo sg voc</td>
<td>21</td>
</tr>
<tr>
<td>wodom pl dat</td>
<td>36</td>
</tr>
<tr>
<td>wody sg gen</td>
<td>6234</td>
</tr>
<tr>
<td>wody pl nom</td>
<td>6336</td>
</tr>
<tr>
<td>wody pl acc</td>
<td>6213</td>
</tr>
<tr>
<td>wody pl voc</td>
<td>6178</td>
</tr>
<tr>
<td>wodzie sg dat</td>
<td>1665</td>
</tr>
<tr>
<td>wodzie sg loc</td>
<td>1585</td>
</tr>
<tr>
<td>wód pl gen</td>
<td>5898</td>
</tr>
</tbody>
</table>

Table 1.1: Results of query \([\text{base}=\text{woda}]\) group by orth, number, case over the IPI PAN Corpus of Polish (in alphabetical order).

## 2 Basic syntax

### 2.1 Simple queries

The pattern matches can be grouped according to a set of segment attributes, specified in grouping rules. The simplest grouping rule consists of one attribute name. For example, to find the frequencies of the forms of the word *woda* (water), one could write:

\([\text{base}=\text{woda}]\) group by orth

The results of the query is a table. Each different value of the specified attribute (orth) encountered in the matches of the first part of the query (\([\text{base}=\text{woda}]\)) corresponds to one row in the results. Each row displays a value of the specified attribute, and the number of matches that contain this particular value.

It is possible to include more attributes in grouping rules, separated by commas, e.g.:

\([\text{base}=\text{woda}]\) group by orth, number, case

In the results of this query (as shown in table 1.1), each row corresponds to a unique combination of values of the specified attributes (*woda* sg nom, *wody* pl nom, *wody* sg gen, etc.) This takes into account a distinction between homonymic forms (*wody* may be sg gen, pl nom, etc.)
2.2 Multiword patterns

Poliqarp patterns can return matches longer than a single segment. To specify the segment whose attribute will be used for grouping, one should add the segment number (with a dot) before the name of the attribute. For example, to find all verbs occurring immediately after the word *woda* (water), one can type:

```
[base=woda][pos=verb] group by 2.base
```

Specification `2.base` refers to the base form of the second segment of the match (the verb after *woda*). Note that subsequent numbers refer to the subsequent segments of the match, not segment specifications in the query. For example, to find the frequency distribution of three subsequent adverbs, the user should type (1. can be skipped):

```
[pos=adv]{3} group by 1.base, 2.base, 3.base
```

To make it possible to address segments in matches of possibly variable length, negative numbers can be used as segment specifications. Such numbers mean counting from the end of the match. For example, to allow an optional adverb between *woda* and the verb, the query should be modified as follows:

```
[base=woda][pos=adv]?[pos=fin] group by -1.base
```

Specification `-1.base` specifies the base form of last segment of the result. Similarly, `-2. ` would refer to the second last, `-3. ` — third last, etc.  

2.3 Ambiguities

One of the features that distinguish Poliqarp from other search tools is the representation and processing of ambiguities. Each segment in a corpus can have a number of interpretations. For example, the Polish word *mam* can be a form of the verb *mieć* (to have) or the noun *mama* (mom). In fact, only *orth* and *length* are attributes of the segment itself — all the other ones (base form, grammatical class, and categories) constitute its interpretation.

By default, one random interpretation of each segment is chosen for grouping. But if `interp combine` is added after an attribute specification, the value of the attribute will be calculated as a concatenation of all the
unique values of the attribute in all the interpretations, separated by a vertical bar. For example, to find all possible interpretations’ combination for word forms that may be a form of the verb mieć, one could write:

\[ \text{[base=mieć] group by base interp combine} \]

The results of such query will include word pairs like *mama/mieć* (mom/to have), *mieć/mienie* (to have/property), *maić/mieć* (to decorate with leaves and flowers/to have), *mielić/mieć* (to grind/to have), etc.\(^4\)

### 2.4 Results sorting and selection

Results can be sorted in alphabetic order (*sort a fronte*) or according to their frequency (*sort by freq*). If partial grouping is used, the results can be also sorted according to a collocation function — see 3.2 for details.

The results selection is now limited to a frequency threshold (*min n*).

### 3 Collocations

#### 3.1 Partial grouping

It is quite easy to find most frequent bigrams using only the basic syntax:

\[ \text{[]} \text{[]} \text{group by 1.base, 2.base sort by freq} \]

However, such results are often insufficient. For example, in collocation detection, not only the bigram frequency, but also the frequencies of its constituents have to be taken into account. Therefore, a special separator — semicolon `;` — has been introduced, which makes it possible to split the grouping rules into two parts. For example:

\[ \text{[]} \text{[]} \text{group by 1.base; 2.base sort by freq} \]

will cause the program to group the results by: 1. *base* (the part before the semicolon); 2. *base* (the part after the semicolon); 1. *base*, 2. *base* (both). For each line of the results of the last grouping, the results of the partial groupings should also be displayed. The *sort* and *min* modifiers are applied to the last grouping.\(^5\)

Each of the grouping parts may include more than one attribute, but the grouping may have no more than two parts. In other words: the grouping rules can include any number of commas, but not more than one semicolon.

The syntax does not necessarily have to be used for bigrams. Different grouping parts may even include references to the same segment, for example:

\[ \text{[base=woda] group by number; case} \]

\(^4\)The results will vary significantly depending on the value of the “Show only disambiguated results” option in Poligarp configuration; if it is checked, the interpretations discarded by the tagger will not appear in the results.

\(^5\)Therefore, the query with partial grouping will return results in the same order as the query with a comma instead of the semicolon.
3.2 Collocation functions

A few dependency measures for statistical detection of collocations have been added as possible sorting parameters (for example sort by cp or sort by dice) in queries with partial grouping. All the currently implemented measures are functions of the following parameters:

\[ c(w_1) \] — number of occurrences of \( w_1 \), where \( w_1 \) is the combination of values of the attributes defined by the first part of the grouping rules;

\[ c(w_2) \] — number of occurrences of \( w_2 \), where \( w_2 \) is the combination of values of the attributes defined by the second part of the grouping rules;

\[ c(w_1w_2) \] — number of occurrences of the combination of \( w_1 \) and \( w_2 \).

The functions are:

\[ \text{cp} \] — conditional probability

\[
\text{cp}(w_1, w_2) = \frac{c(w_1w_2)}{c(w_1)}
\]

\[ \text{scp} \] — symmetric conditional probability

\[
\text{scp}(w_1, w_2) = \frac{c(w_1w_2)^2}{c(w_1)c(w_2)}
\]

\[ \text{maxcp} \] — maximum conditional probability

\[
\text{maxcp}(w_1, w_2) = \max\left( \frac{c(w_1w_2)}{c(w_1)}, \frac{c(w_1w_2)}{c(w_2)} \right)
\]

\[ \text{dice} \] — Dice’s formula

\[
\text{dice}(w_1, w_2) = \frac{2c(w_1w_2)}{c(w_1) + c(w_2)}
\]

Because dependency measures in fact prefer rare bigrams, a minimum frequency threshold is recommended, for example:

\[ [\text{pos}\neq\text{interp}]\{2\} \text{ group by base; 2.base sort by scp min 2} \]

An alternative approach is to add some frequency bias to the dependency test value, for example:

\[ [\text{pos}\neq\text{interp}]\{2\} \text{ group by base; 2.base sort by scp bias 0.5} \]

\( \text{bias} \) \( b \) means “before sorting, multiply the function results by power \( b \) of frequency”:

\[ x \text{ bias } b = x \cdot c(w_1w_2)^b \]
Aleksander Buczyński

For example:

\[ scp \text{ bias} 0.5 = \frac{c(w_1w_2)^{2.5}}{c(w_1)c(w_2)} \]

Of course, bias and min keywords can be combined,\(^6\) for example:

\[ \{2\} \text{ group by base; 2. base sort by scp bias 0.5 min 2} \]

4 Syntax synopsis

A statistical query has the following syntax (square brackets denote optional parts):

\(<\text{pattern}> \text{ group by } <\text{attr list}> [; <\text{attr list}>] [\text{interp } <\text{method}>] [\text{sort } <\text{order}>] [\text{min } <\text{cmin}>] \]

where:

- \(<\text{pattern}>\) is a Poliqarp query; only segment sequences matching \(<\text{pattern}>\) will be taken into account in the statistics;
- \(<\text{attr list}>\) is a list of attribute specifications (for example base or 2.case), separated by commas; each attribute specification consists of an optional segment specification (for example 2. or -1.), and an obligatory attribute name (for example base or case);
- \(<\text{method}>\) is an interpretation selection method (random or combine);
- \(<\text{order}>\) is a sorting order, as described in 2.4 (simple queries) and 3.2 (queries with partial grouping);
- \(<\text{cmin}>\) is a minimum frequency threshold; only results which occurred at least \(<\text{cmin}>\) times in the matches should be displayed.

5 Conclusion

A client application of Poliqarp understanding the described syntax (i.e., the pattern matching is handled by the Poliqarp server, and the grouping — by the client) has been implemented and is currently in beta testing.

The syntax seems to be very flexible and able to cover many different linguistic queries. It was also relatively easy to implement on top of the existing search engine. On the other hand, the flexibility makes it difficult for the implementation to compete in efficiency with more specialised tools, for example for collocation detection. The utility is best suited for quick preliminary testing of linguistic hypotheses on small samples of data, or researching relatively rare phenomena (at least not extremely common) — up to a few hundred thousand occurrences in a corpus.

\(^6\)For consistency, the minimum threshold is still applied to the bare frequency, not the biased collocation function.
Bibliography


No future Adams pairs: applying the
global/local conditional probability
distinction

KEVIN DEMIDDELE
University of Leuven
kevin.demiddele@hiw.kuleuven.be

ABSTRACT. The distinction between global and local probability, introduced by Stefan Kaufmann, is used to analyze an example by Adam Morton. With it, Morton aims to show that there are future Adams pairs. It is argued here that Morton’s argument is unconvincing. The global-local probability distinction is also used to analyze an alleged counterexample to modus ponens by Gillies. The expectation is that the distinction will be useful for resolving some other puzzles involving context-dependent conditionals.

1 Morton’s example

Adam Morton [10] claims to show that there are Adams pairs set in the future. Regular Adams pairs mark the semantic difference between indicative and subjunctive conditionals. Let me use Morton’s example to illustrate this: most people would readily accept (1.1), while almost no one would be willing to accept (1.2).

If Shakespeare did not write Hamlet, someone else did. (1.1)

If Shakespeare had not written Hamlet, someone else would have. (1.2)

It is more or less generally accepted that this reflects a semantic distinction between two classes of conditionals. But a much debated topic is whether future conditionals belong in the indicative or the subjunctive class.1 In the context of this discussion, Morton wants to show that there are also Adams pairs set in the future, with the following example. In the distance, we see Lara standing next to a bomb. We know that most of the bombs are dangerous, so we accept (1.3).

If Lara touches the bomb, it will explode. (1.3)

1Traditionally, they were considered to be indicative. An early attempt to show why they should be considered subjunctive is [4].
On the other hand, we also know that Lara can see whether the bomb is dangerous or not (but we cannot see it from here) and that she will only touch it if it is not dangerous, so we do not accept (1.3) after all.

According to Morton, (1.3) has an ‘indicative’ and a ‘subjunctive’ meaning, which happen to be expressed in the same sentence. We accept the conditional in (1.3) as a subjunctive, but not as an indicative. So we have in fact a future tense Adams pair. To exhibit it more clearly as a pair, consider (1.4): we accept this conditional as well, in its indicative reading, although it seems to be incompatible with (1.3).

If Lara touches the bomb, it will not explode. \hspace{1cm} \text{(1.4)}

The first conditional, says Morton, has a ‘subjunctive’ meaning and could be restated as ‘It will be the case that if she had touched it, it would have exploded’. The second one has an ‘indicative’ meaning and could be restated as ‘It will be the case that if she did touch it, it didn’t explode’.

I disagree with Morton that the different interpretations of (1.3) and (1.4) constitute an Adams pair. A key difference lies in the fact that with genuine Adams pairs it is not unreasonable to accept both the conditional with the positive consequent and that with the negated consequent simultaneously. This is not the case with (1.3) and (1.4). I accept either the first, if I focus on the information that most bombs are dangerous, or I accept the second, focussing on the information that Lara is smart enough to keep her hands off of dangerous bombs. No such shift in background information is needed to accept (1.1) but not (1.2). These conditionals have different meanings. If they could be formulated with the same words, there would be genuine ambiguity. If someone said ‘If not Shakespeare, then someone else’ and it is not clear whether he intends this as an indicative or as a subjunctive conditional, you cannot agree or disagree with that person. You need to know what he means. On the other hand, ‘What do you mean?’ does not seem like an acceptable response to someone telling you ‘If Lara touches the bomb, it will explode’.

The reasoning behind the acceptance of both conditionals is clear. We do not know whether the specific bomb is dangerous or not, otherwise the situation would be very simple: if the bomb was dangerous, (1.3) would be acceptable and (1.4) would not; if the bomb was not dangerous, it would be the other way around. But although we do not know whether this specific bomb is dangerous, we do have some information about it and about Lara. We know that the bomb is most likely dangerous and that dangerous bombs explode when touched, which provides evidence for accepting (1.3). We also know that Lara does not touch dangerous bombs, which provides evidence for accepting (1.4). Still, these basic intuitions do not suffice to support Morton’s claim that both conditionals can be consistently accepted simultaneously. For, even with the limited amount of information we have, the
correct values of belief to be attributed to (1.3) and (1.4) can be calculated, as will now be shown.

\section{Local and global conditional probability}

Morton’s example is not an Adams pair, but an example of the distinction between global and local probability, introduced by Kaufmann \[7\]. These are two different ways of calculating the probability of a conditional ‘$A \rightarrow C$’, in case there is a third variable $X$ on which the probability of the consequent depends, and that is itself causally independent of, but stochastically dependent upon the antecedent. Kaufmann introduced the distinction as a systematic way to analyze counterexamples against the thesis that the probability of a conditional is its conditional probability. Theories endorsing this thesis have their origin in \[1\] and the point of view of this paper is one of agreement with the thesis.

This is how local and, respectively, global probability are defined: \(^2\)

\[
\begin{align*}
p_l(A \rightarrow C) &= p(C | A \land X)p(X) + p(C | A \land \neg X)p(\neg X) \\
p_g(A \rightarrow C) &= p(C | A \land X)p(X | A) + p(C | A \land \neg X)p(\neg X | A) \quad (1.5)
\end{align*}
\]

The difference is that in the global probability, the antecedent affects the weights of the conditional probabilities, while in the local probability, it does not. The global probability is the standard conditional probability; the probability of the consequent, given the antecedent. Kaufmann’s point \[7, p.594\] is that this probability sometimes differs from the local probability, which he takes to be the prior probability of the conditional.

The distinction between global and local conditional probability can be applied to develop a better understanding of Morton’s example. Take $T$ for ‘Lara touches the bomb’ and $E$ for ‘the bomb explodes’. The third variable here is $D$, ‘The bomb is dangerous’. Most bombs are dangerous, if a dangerous bomb is touched it explodes and Lara will not touch a dangerous bomb, so take $p(D) = .9$, $p(E | T \land D) = .99$, $p(D | T) = .01$, $p(E | T \land \neg D) = 0$. Of course, the precise value of these probabilities does not matter, as long as they correspond to the intuitive values like ‘high’ and ‘low’ the example provides us with. The difference between global and local probability now amounts to taking or not taking into account the information that it is less probable that the bomb is dangerous if Lara touches it. If we do not take this into account, (1.6) shows how we obtain the high value of the local probability and accept (1.3). If we do take this information into account, the global probability gives us a low value, as in (1.7), so we do not accept

---

\(^2\)I give the simplified definitions where the variable $X$ takes only two different values that jointly exhaust the probability space. This will do for the application to Morton’s example.
No future Adams pairs: applying the global/local conditional probability distinction

\[(1.3)\]
\[
\begin{align*}
p_l(T \rightarrow E) &= p(E|T \land D)p(D) + p(E|T \land \neg D)p(\neg D) \\
&= 0.99 \times 0.9 + 0 \times 0.1 = 0.891
\end{align*}
\]

\[\begin{align*}
p_g(T \rightarrow E) &= p(E|T \land D)p(D|T) + p(E|T \land \neg D)p(\neg D|T) \\
&= 0.99 \times 0.01 + 0 \times 0.99 = 0.0099
\end{align*}\]  

Kaufmann, too, points out [7, p.588] that you can convince yourself that the global probability is low, and that you can convince yourself that the local probability is high, but not both simultaneously. Although global and local probability explain how both (1.3) and (1.4) can seem intuitively acceptable, it is clear that a rational agent should take into account all the information that is available to him, so in our scenario, (1.4) is acceptable, while (1.3) is not.

Furthermore, the distinction between global and local probability also applies to past tense indicatives and to subjunctives. With the same shift of attention from the information that \(p(D)\) is high to the information that \(p(D|T)\) is low, we can also accept both conditionals in (1.8) and those in (1.9).

If Lara touched the bomb, it exploded.
If Lara touched the bomb, it did not explode.  

If Lara had touched the bomb, it would have exploded.
If Lara had touched the bomb, it would not have exploded.

It is clear then, that the distinction does not mark the difference between an indicative and a subjunctive meaning, as Morton claims. The distinction was supposed to be analogous with the one between indicative and subjunctive meaning, but once it is shown that the distinction between global and local probability applies across the future-past and indicative-subjunctive distinctions, that analogy breaks down.

3 Back-tracking

If we asked a person, with beliefs that are in concordance with Morton’s example, whether the bomb will explode if Lara touches it, the following answer does not seem to be acceptable: ‘Actually, both. If she touches it, it will explode, because it is dangerous. And if she touches it, it will not explode, because she would not do anything stupid.’ This, as said before, is in contrast with genuine Adams pairs. Moreover, Morton’s way of putting it does not seem to be an answer to the question: ‘It will be the case that if she had touched it, it would have exploded.’ He notes that it is a cumbersome way of saying it, but if people are pressed to make clear what they
mean, as here, cumbersome expressions are not exceptional, so there must be another reason why it is not acceptable. The best explanation seems to be that there is only one meaning, not two. What may go through a person’s mind when being asked whether the bomb will explode if Lara touches it, is plausibly something along the following lines: ‘Most likely, that bomb is dangerous. So if Lara touches it, it will explode. But wait: Lara will not make a mistake. The only reason why she would touch the bomb is because she has determined it to be harmless. So if she touches it, it will not explode after all.’ This kind of argument shows great similarity to what has been called a back-tracking argument in the literature on subjunctive conditionals.\(^3\) The literature on back-tracking subjunctives is vast and cannot be dealt with in great detail here. Back-tracking subjunctives have a ‘reverse’ counterfactual dependence: events at a time \(t_1\) depend upon events at a later time \(t_2\). ‘If Lara were to touch the bomb, the bomb would have to have been harmless’ is an example. Back-tracking subjunctives have given rise to great complications and it is a matter of discussion whether theories of subjunctives should allow for back-tracking or not. Suffice it here to note that back-tracking seems less problematic with indicatives. The trouble with back-tracking subjunctives comes from ‘mingling causal inference with appeal to actual fact’\(^3\), p.208. But whereas subjunctives have to do with causal powers, indicatives have to do with belief revision. Changing one’s mind seems less puzzling than changing (reverse) causal connections. According to Bennett \(3\), p.274\) the essential point about these conditionals is that these are cases where the consequent is the best explanation for the antecedent and only for subjunctives, but not for indicatives, does this entail back-tracking. The modal element, that the consequent ‘must’ be the case, is retained. In our example, the consequent is a consequence of what explains the antecedent. ‘If Lara touches the bomb, the bomb must be harmless, and therefore it does not explode.’ This is what Bennett calls a V-shaped explanation\(3\), p.339\).

Morton is of course right that we can have subjunctive thoughts about the future. But it is incorrect to express them using the same future conditional sentence we use to express our indicative thoughts about the future. We endorse the subjunctive thought only in case Lara does not touch the bomb, so we can express it as follows: ‘She will not touch it and it will be the case that if she had touched it, it would have exploded.’ The subjunctive ‘It will be the case that if she had touched it, it would have exploded’ turns out to be false if the bomb is not dangerous and Lara safely touches it. Or more appropriately: the subjunctive form turns out to be infelicitous if the antecedent is true, and the corresponding indicative turns out to be false. If it is allowed to have opposing beliefs in a future indicative conditional

\(^3\)The locus classicus is \(8\). Kaufmann\(7\), p.603\) handles a case like this and claims that the global-local probability distinction can be applied to it.
and its corresponding subjunctive, one of the main arguments for reclassifying future indicatives in the class of subjunctives is undermined. This would be very strange for Morton’s argument, which is aimed at rebutting an argument by Bennett against the reclassification of future indicatives.

The indicative thought is simply expressed by ‘If she touches it, it will explode’. But this we accept only if she does touch it, i.e. if she judges the bomb to be harmless. So the complete thought we endorse is this: ‘Either she will not touch it and it will be the case that if she had touched it, it would have exploded or she will touch it and it will not explode.’ This is an exclusive disjunction. Either she will touch the bomb or not. It is inconsistent to accept both conditionals in Morton’s alleged Adams pair.

Note that I refrain from formulating the indicative part this way: ‘She will touch it and it will be the case that if she did touch it, it didn’t explode.’ The fact that Morton talks about conditionals being the case in fact commits him to a specific strand of theories about conditionals. These theories are opposed to the so called ‘suppositional view’ of conditionals, in which conditionals are not regarded as categorical statements, i.e. they are not things that can be ‘the case’. Since my critique on Morton’s point rests on Kaufmann’s defence of a suppositional account of conditionals, this discussion tends to boil down to fundamental opinions about the semantics of conditionals. I will not go into this here.

4 Comparison to Gibbard’s example

Several referees suggested me to compare Morton’s example to Gibbard’s famous Sly-Pete example. There are indeed some notable similarities which make a comparison worthwhile, but there are also important differences and it does not seem to be a case that can be explained using the distinction between global and local probability.

Gibbard [5] defends Adams’ non-truthconditional account of conditionals against truthconditional accounts, and particularly against that of Stalnaker. In doing so, he presented some examples that became classics in the field. One of these is the ‘Sly Pete story’. Gibbard observes that non-truthfunctional theories that treat conditionals as propositions all adhere to the law of Conditional Non-contradiction, which says that $A \rightarrow C$ is inconsistent with $A \rightarrow \neg C$. The Sly Pete story is then presented as a case which becomes problematic for these theories to explain:

Sly Pete and Mr. Stone are playing poker on a Mississippi riverboat. It is now up to Pete to call or fold. My henchman Zack sees Stone’s hand, which is quite good, and signals its content to
Kevin Demiddele

Pete. My henchman Jack sees both hands, and sees that Pete’s hand is rather low, so that Stone’s is the winning hand. At this point, the room is cleared. A few minutes later, Zack slips me a note which says ‘If Pete called, he won’ and Jack slips me a note which says ‘If Pete called, he lost.’ I know that these notes both come from my trusted henchmen, but do not know which of them sent which note. I conclude that Pete folded.[5, p.231]

The point is that neither Zack nor Jack have any false beliefs and so they both rightfully assert their respective conditional. Gibbard concludes ‘Neither, then, could sincerely be asserting anything false. Each is sincere, and so each, if he is asserting a proposition at all, is asserting a true proposition.[5, p.231]’

According to Slater [11, p.436] the probabilistic solution to Gibbard’s example has been suppressed. He contends that no one has given enough attention to the qualification ‘most probably’. The fact of the matter is then simply that, when one person is entitled to accept ‘$A \rightarrow C$’ while another is entitled to accept ‘$A \rightarrow \neg C$’, their evidence is not equally conclusive. In the original example by Gibbard, Zack is only entitled to assert ‘Most probably, if Pete called, he won.’ Jack is entitled to utter the conditional without the ‘probably’ qualification. In apparent contrast with this, Bennett[3, p.83] gives a similar example in which both persons are ‘fully entitled [to accept their conditional]; these acceptances are intellectually perfect’. But Bennett adapts Gibbard’s example. He does this, because, in the original example, the evidence for the conditional ‘If Pete called, he lost’ is also evidence for the conditionals ‘If Pete were to have called, he would have lost’ and ‘If Pete had called he would have lost’ and Bennett believes that the discussion about subjunctives complicates the discussion on the Gibbard case. By changing the example to a case where the evidence only supports the indicative, this is fixed. But it also means that in Bennett’s example, no one is entitled to believe his conditional with full certainty.

Gibbard’s example bears some resemblance to the example we have treated earlier because it also involves rightful acceptance of conflicting conditionals. But there are important differences. First, Gibbard’s example involves two persons. Whereas Morton claimed the two conflicting conditionals to be acceptable by the same belief set, i.e. a probability distribution, Gibbard’s example involves two different probability distributions. Both persons in the story do not attribute high probability to false propositions, but since one has more information than the other, it is less puzzling to understand how they can arrive at conflicting beliefs about the conditional. Second, Gibbard’s example does not seem to involve the indicative-subjunctive distinction. This is made clear by Bennett’s adaptation, which completely unties it from the realm of subjunctives. Third, the example does not involve a third, causally independent but stochastically dependent,
variable and this is the reason why the distinction between local and global conditional probability is not the key to understanding it.

5 A second application: counterexamples to modus ponens

The inapplicability of the distinction to the apparently similar example by Gibbard does not mean that the use of the distinction to explain Morton’s example was ad hoc. It can be expected that local probability lies at the basis of many other fallacious arguments that involve conditionals. This was already convincingly shown by Kaufmann himself [7]. This will be further illustrated by applying it to a counterexample to modus ponens given by Gillies[6, p.592]. Similar counterexamples were presented earlier by Vann McGee[9].

Gillies’ counterexample is as follows: there has been a murder in the mansion. There are three suspects: the driver, the gardener and the butler. The first two belong to the grounds staff, the latter one to the house staff. The information we have is that it is almost certainly the gardener who did it (G), it might be the butler (B), and it is almost certainly not the driver (D). So we know (1.10) and since we strongly suspect the gardener, we are quite sure about (1.11).

If a member of the grounds staff is the culprit, then if it is not the gardener who is guilty, the driver is. (1.10)

A member of the grounds staff is the culprit. (1.11)

By modus ponens, these two premisses yield (1.12), which is not consistent with our information.

If it is not the gardener who is guilty, the driver is. (1.12)

If it is not the gardener, our main suspect is the butler, not the driver.

The trick is that you accept the premise (1.11) on the basis of your high probability for G, while the conclusion (1.12) talks about a not-G situation. So somehow we need to remember our acceptance of that premise. In Kaufmann’s terms, B is the third variable, which is dependent on the antecedent of (1.12). If we ‘forget’ this, the local probability yields the counterintuitive

---

5 As a referee pointed out, it is more accurate and more in line with intuitions to take the third variable to be ‘It is not someone from the grounds staff’. In the story, this is equivalent to ‘The butler did it’. Since the letter G was already used expressing the proposition ‘The gardener did it’, I retained the letter B. Throughout the example, B can be read as ‘It is not someone from the grounds staff’.

62
result. Expressing our information in probabilities, let us say \( p(G) = .70, \) \( p(B) = .25, \) \( p(D) = .05. \)

\[
p_l(\neg G \rightarrow D) = p(D|\neg G \land B) \times p(B) + p(D|\neg G \land \neg B) \times p(\neg B)
\]

\[
= 0 \times 0.25 + 1 \times 0.75 = 0.75
\]

(1.13)

If we, correctly, calculate the global probability, the oddity disappears:

\[
p_g(\neg G \rightarrow D) = p(D|\neg G \land B) \times p(B|\neg G) + p(D|\neg G \land \neg B) \times p(\neg B|\neg G)
\]

\[
= 0 \times 0.833 + 1 \times 0.166 = 0.166
\]

(1.14)

6 Conclusion

We have seen how the distinction between local and global conditional probability enables us to obtain a better understanding of Morton’s complicated example. As has been extensively argued for, this example does not provide evidence for the claim that there exist future Adams pairs. Local probability may prove to be no more than a fallacy,\(^6\) but it does explain how some people’s intuitions go wrong in judging rather complicated conditionals, or why some wrongly believe that there are future Adams pairs. We also used the distinction to analyze an alleged counterexample to modus ponens. The expectation is that the distinction will be useful for resolving some other puzzles involving context-dependent conditionals.\(^7\)

Bibliography


\(^6\)Kaufmann seems to think it is not just a fallacy. Whether he is right or not is not the issue here.

\(^7\)I would like to thank Igor Douven, Jan Heylen and Leon Horsten for useful comments on a previous version of this paper. I would also like to thank the anonymous ESSLLI Student Session referees for their very helpful and interesting remarks.


Memory-Based Word Sense Disambiguation for Romanian with Automatic Feature Selection

GEORGIANA DINU
Seminar für Sprachwissenschaft
Universität Tübingen, Germany
gdinu@sfs.uni-tuebingen.de

ABSTRACT. This paper describes a memory-based approach to the Romanian lexical sample Word Sense Disambiguation task which was part of the SENSEVAL-3 evaluation exercise. The system employs a total set of very simple contextual features and a per-word feature selection algorithm. The overall accuracy is higher than the accuracy of the best-performing system participating in the task. The result is very promising, considering the fact that both in terms of algorithms used and in terms of methods of extracting features the complexity of these systems is much higher.

1 Introduction

Word sense disambiguation (WSD) consists in automatically assigning senses to occurrences of polysemous words. The SENSEVAL-3 WSD evaluation exercise provided data for Romanian consisting of 39 ambiguous words.

In building a supervised WSD system the main decisions consist in the choice of a classifier, the choice of features used as indicators for a word’s sense and algorithms for optimally determining and combining the two. Memory-based learning (MBL) is a supervised learning method that has been previously successfully used in Word Sense Disambiguation (e.g. (Ng and Lee 1996), (Veenstra, den Bosch, V., Buchholz, Daelemans, and Zavrel 2000), (Mihalcea 2002)). The system described in this paper defines a set of simple contextual features to be used in building a classifier with TiMBL (Daelemans, Zavrel, van der Sloot, and van den Bosch 1999) and implements an automatic per-word feature selection algorithm. The feature selection algorithm follows closely the one described in (Mihalcea 2002), which has been shown to be quite performant on English SENSEVAL-2 data. For each word an optimal feature configuration is built by adding features one by one, after having determined, at each step, which one brings the best improvement.
Section 2 describes in more detail how a feature space and feature combinations are chosen for the Romanian disambiguation task. Section 3 reports best results as well as those obtained on other system configurations followed by a discussion in Section 4.

2 Main Algorithm

All experiments are run on SENSEVAL-3 Romanian lexical sample data, which consists of labeled examples for 39 ambiguous words: 25 nouns, 9 verbs, and 5 adjectives. The senses, with an average of 8.8 senses per word fine-grained (4.7 coarse-grained) are manually extracted from a Romanian dictionary (Coteanu, Seche, Seche, Burnei, Ciobanu, Contras, Creţa, Hristea, Mares, Stingaciu, Stănescu, Tugulea, Vulpescu, and Hristea 1975). The annotated data is a part of RoCo corpus, a collection of Romanian newspaper text and it has been tokenized and part-of-speech tagged using RACAI tools (Tufis 1999). The tagging is estimated to be 98% accurate.

Automatic feature selection consists in determining an optimal feature subset, given a set of features which contains all those that might be considered helpful. However different words have different indicators for their senses, and their disambiguation may profit in different ways, sometimes opposite, from the use of a particular feature. Also many words are disambiguated with high accuracy with the use of very restricted context information (for example the preceding word) while others need all the information they can be provided with. Including uninformative features in building the training data for a word affects the performance of MBL classification, even if small weights are assigned to those features. Building per-word feature configurations is therefore justified in order to adapt the disambiguation process to the word’s specific requirements.

2.1 Selection algorithm

A pool of features, containing all the features is initially generated. For each word a good feature combination is afterwards determined in the following way: features are added one by one, for as long as adding features gives any improvements. The decision whether to add a feature is taken according to how much it improves the accuracy of a cross-validation, as output by TiMBL, when added to the current feature set. Unlike an exhaustive search for the optimal feature configuration, for which the number of runs grows exponentially with the size of the feature space, this algorithm is very efficient. For the 20 feature space considered in the experiments described, the average number of TiMBL runs necessary for determining the setting for a word is approximately 100, vs $2^{20}$ necessary for determining the absolute optimal set.

Similar to (Mihalcea 2002), the algorithm is summarized below:
generate a pool of features PF

for each word

initialize the set of selected features $SF = \emptyset$

repeat

for each $F \in PF$

run a cross validation run on training data using the set of features $SF \cup \{F\}$

determine $F$ for which $SF \cup \{F\}$ lead to the best accuracy

add $F$ to the set of selected features $SF$

remove $F$ from the pool of features $PF$

until no improvement

return $SF$

The low computation cost of the algorithm makes it possible to extend the search for the best feature configuration. This can be done by stopping the selection algorithm for one word only when adding any of the remaining features determines a decrease in accuracy or by allowing a maximum accuracy decrease threshold.

2.2 Feature space

From each training or test instance a feature vector containing a number of maximum 20 feature values can be extracted. The features are chosen to be as simple as possible, but they are among those reported in the literature as best-performing. They are extracted from a local context around the ambiguous word. This context is provided with part-of-speech annotation but is not processed in any other way (the usual lemmatization or removal of stop words is not performed). Larger context is ignored and sense specific keywords are not used. Instead the ambiguous word itself, as it appears in the context, is used as a sense indicator. The following list describes the features, preceded by abbreviations used in reporting the results.

- CT$_k$ (Context token) This feature considers the token situated at position $k$ relative to the target word. Parameter $k$ is in the [-3..3] range. CT$_0$ is the word form of the ambiguous word.

- CP$_k$ (Context POS) The part-of-speech tag of the token situated at position $k$ relative to the target word. Parameter $k$ is in the [-3..3] range. CP$_0$ is the part-of-speech tag of the ambiguous word.

- VA (Verb after) The first verb found after the target word.

- VB (Verb before) The first verb found before the target word.
• NA (Noun after) The first noun found after the ambiguous word.
• NB (Noun before) The first noun found before the ambiguous word.
• PA (Preposition after) The first preposition found after the ambiguous word.
• PB (Preposition before) The first preposition found before the ambiguous word.

All the features are extracted from the narrow context of the target word: if a sentence barrier is crossed, the corresponding feature will be assigned a constant ‘null’ value.

3 Experiments and results

The first experiment consisted in determining an optimal feature set for each word, according to the feature selection algorithm described in section 2.1.

Both on training data and on test data TiMBL was used with default settings: IB1 algorithm, Gain Ratio weighting and \( k = 1 \). For evaluating the performance of a feature configuration, leave-one-out cross-validation was performed. After the selection has been performed for each word, the corresponding feature vectors are extracted from train and final test data.\(^1\) TiMBL is run on this data and a prediction file is created. Afterwards the scores are computed using the official scoring software, which provides both fine-grained and coarse-grained accuracies. Tables 1.1, 1.2 and 1.3 show the results obtained for the nouns, verbs and adjectives included in the task. For each word the tables include an approximate translation of their most common sense. The test data size is omitted as it consists of a number of instances approximately equal to half of the training size. Both fine-grained and coarse-grained evaluation scores are given. The baseline reported is computed by assigning the most frequent sense (MFS) to the test instances and running it through the scoring software.

Compared to the MFS baseline, nouns achieve a net gain of 12.1% (14.4% coarse-grained) and verbs 18.4% (16.4% coarse). Adjectives are disambiguated best for the Romanian task, achieving an accuracy gain of 28.7% (23% coarse). The error reduction rates for fine-grained scores are 33.4% for nouns, 40% verbs and 46.3% for adjectives.

The overall results are better than those of the systems participating in the Romanian task, the most accurate one being outperformed by 1.3% in fine-grained evaluation and 1.6% in coarse-grained.

A second experiment aimed at evaluating how much feature selection influenced the overall performance. TiMBL was used again, but this time

\(^1\)The train/test split is identical to the ones provided in SENSEVAL
Table 1.1: MBL with per-word feature selection. Nouns

<table>
<thead>
<tr>
<th>word</th>
<th>translation</th>
<th>size</th>
<th>MFS (f)</th>
<th>MFS (c)</th>
<th>acc. (f)</th>
<th>acc. (c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ac</td>
<td>needle</td>
<td>127</td>
<td>50.8</td>
<td>50.8</td>
<td>73.8</td>
<td>75.4</td>
</tr>
<tr>
<td>accent</td>
<td>accent</td>
<td>172</td>
<td>73.6</td>
<td>77.0</td>
<td>89.7</td>
<td>93.1</td>
</tr>
<tr>
<td>actiune</td>
<td>action</td>
<td>261</td>
<td>39.8</td>
<td>39.8</td>
<td>61.7</td>
<td>85.2</td>
</tr>
<tr>
<td>canal</td>
<td>channel</td>
<td>134</td>
<td>68.2</td>
<td>68.2</td>
<td>69.7</td>
<td>75.8</td>
</tr>
<tr>
<td>circuit</td>
<td>circuit</td>
<td>200</td>
<td>49.5</td>
<td>50.5</td>
<td>59.4</td>
<td>65.3</td>
</tr>
<tr>
<td>circulatie</td>
<td>circulation</td>
<td>221</td>
<td>45.6</td>
<td>45.6</td>
<td>59.4</td>
<td>68.4</td>
</tr>
<tr>
<td>coroana</td>
<td>crown</td>
<td>252</td>
<td>58.7</td>
<td>61.9</td>
<td>77.0</td>
<td>77.8</td>
</tr>
<tr>
<td>delfin</td>
<td>dolphin</td>
<td>31</td>
<td>100</td>
<td>100</td>
<td>80.0</td>
<td>80.0</td>
</tr>
<tr>
<td>demonstratie</td>
<td>demonstration</td>
<td>229</td>
<td>64.3</td>
<td>64.3</td>
<td>73.0</td>
<td>73.0</td>
</tr>
<tr>
<td>eruptie</td>
<td>eruption</td>
<td>54</td>
<td>40.7</td>
<td>40.7</td>
<td>81.5</td>
<td>81.5</td>
</tr>
<tr>
<td>genius</td>
<td>genius</td>
<td>106</td>
<td>72.2</td>
<td>77.8</td>
<td>64.8</td>
<td>70.4</td>
</tr>
<tr>
<td>nucleu</td>
<td>nucleus</td>
<td>64</td>
<td>78.8</td>
<td>78.8</td>
<td>81.8</td>
<td>81.8</td>
</tr>
<tr>
<td>opozitie</td>
<td>opposition</td>
<td>266</td>
<td>96.3</td>
<td>96.3</td>
<td>95.5</td>
<td>95.5</td>
</tr>
<tr>
<td>perie</td>
<td>brush</td>
<td>46</td>
<td>79.2</td>
<td>95.8</td>
<td>75.0</td>
<td>95.8</td>
</tr>
<tr>
<td>pictura</td>
<td>painting</td>
<td>221</td>
<td>47.7</td>
<td>47.7</td>
<td>75.7</td>
<td>81.1</td>
</tr>
<tr>
<td>platforma</td>
<td>platform</td>
<td>226</td>
<td>38.8</td>
<td>38.8</td>
<td>58.6</td>
<td>58.6</td>
</tr>
<tr>
<td>port</td>
<td>port</td>
<td>219</td>
<td>51.9</td>
<td>51.9</td>
<td>81.5</td>
<td>83.3</td>
</tr>
<tr>
<td>problema</td>
<td>problem</td>
<td>262</td>
<td>44.3</td>
<td>44.3</td>
<td>69.5</td>
<td>69.5</td>
</tr>
<tr>
<td>proces</td>
<td>process</td>
<td>166</td>
<td>62.2</td>
<td>64.6</td>
<td>81.7</td>
<td>82.9</td>
</tr>
<tr>
<td>reactie</td>
<td>reaction</td>
<td>261</td>
<td>83.2</td>
<td>83.2</td>
<td>83.2</td>
<td>83.2</td>
</tr>
<tr>
<td>stil</td>
<td>style</td>
<td>199</td>
<td>60.4</td>
<td>80.2</td>
<td>62.4</td>
<td>76.2</td>
</tr>
<tr>
<td>timbru</td>
<td>stamp</td>
<td>231</td>
<td>94.0</td>
<td>99.1</td>
<td>94.8</td>
<td>98.3</td>
</tr>
<tr>
<td>tip</td>
<td>type</td>
<td>263</td>
<td>76.3</td>
<td>76.3</td>
<td>87.8</td>
<td>89.3</td>
</tr>
<tr>
<td>val</td>
<td>wave</td>
<td>242</td>
<td>85.1</td>
<td>85.1</td>
<td>87.6</td>
<td>88.4</td>
</tr>
<tr>
<td>valoare</td>
<td>value</td>
<td>251</td>
<td>63.2</td>
<td>75.2</td>
<td>72.8</td>
<td>85.6</td>
</tr>
<tr>
<td>total</td>
<td></td>
<td>-</td>
<td>-</td>
<td>63.8</td>
<td>66.2</td>
<td>75.9</td>
</tr>
</tbody>
</table>

Table 1.2: MBL with per-word feature selection. Verbs

<table>
<thead>
<tr>
<th>word</th>
<th>translation</th>
<th>size</th>
<th>MFS (f)</th>
<th>MFS (c)</th>
<th>acc. (f)</th>
<th>acc. (c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>castiga</td>
<td>win</td>
<td>227</td>
<td>52.2</td>
<td>52.2</td>
<td>72.2</td>
<td>72.2</td>
</tr>
<tr>
<td>citi</td>
<td>read</td>
<td>259</td>
<td>82.3</td>
<td>90.8</td>
<td>82.3</td>
<td>89.2</td>
</tr>
<tr>
<td>cobori</td>
<td>descend</td>
<td>252</td>
<td>47.7</td>
<td>75.8</td>
<td>68.0</td>
<td>85.2</td>
</tr>
<tr>
<td>conduce</td>
<td>drive</td>
<td>265</td>
<td>55.2</td>
<td>56.0</td>
<td>81.3</td>
<td>82.1</td>
</tr>
<tr>
<td>creste</td>
<td>grow</td>
<td>209</td>
<td>43.7</td>
<td>43.7</td>
<td>72.8</td>
<td>74.8</td>
</tr>
<tr>
<td>desena</td>
<td>draw</td>
<td>54</td>
<td>81.5</td>
<td>81.5</td>
<td>81.5</td>
<td>81.5</td>
</tr>
<tr>
<td>desface</td>
<td>untie</td>
<td>115</td>
<td>27.6</td>
<td>32.8</td>
<td>53.4</td>
<td>56.9</td>
</tr>
<tr>
<td>fierbe</td>
<td>boil</td>
<td>83</td>
<td>32.6</td>
<td>37.2</td>
<td>48.8</td>
<td>58.1</td>
</tr>
<tr>
<td>induci</td>
<td>sweeten</td>
<td>19</td>
<td>40.0</td>
<td>80.0</td>
<td>60.0</td>
<td>80.0</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>-</td>
<td>53.9</td>
<td>61.5</td>
<td>72.3</td>
<td>77.9</td>
</tr>
</tbody>
</table>
Table 1.3: MBL with per-word feature selection. Adjectives

<table>
<thead>
<tr>
<th>word</th>
<th>translation</th>
<th>size</th>
<th>MFS (f)</th>
<th>MFS (c)</th>
<th>acc. (f)</th>
<th>acc. (c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>incet</td>
<td>slow</td>
<td>224</td>
<td>41.6</td>
<td>41.6</td>
<td>79.6</td>
<td>79.6</td>
</tr>
<tr>
<td>natural</td>
<td>natural</td>
<td>242</td>
<td>23.6</td>
<td>51.2</td>
<td>67.5</td>
<td>74.8</td>
</tr>
<tr>
<td>neted</td>
<td>smooth</td>
<td>34</td>
<td>41.2</td>
<td>52.9</td>
<td>41.2</td>
<td>41.2</td>
</tr>
<tr>
<td>ofical</td>
<td>official</td>
<td>185</td>
<td>53.1</td>
<td>72.9</td>
<td>72.9</td>
<td>72.9</td>
</tr>
<tr>
<td>simplu</td>
<td>simple</td>
<td>153</td>
<td>36.6</td>
<td>46.3</td>
<td>48.8</td>
<td></td>
</tr>
<tr>
<td>total</td>
<td>-</td>
<td>-</td>
<td>38.1</td>
<td></td>
<td>66.8</td>
<td>69.4</td>
</tr>
</tbody>
</table>

Table 1.4: System results on SENSEVAL-3 Romanian data

<table>
<thead>
<tr>
<th>System</th>
<th>Fine</th>
<th>Coarse</th>
</tr>
</thead>
<tbody>
<tr>
<td>feat_sel MBL</td>
<td>74.0</td>
<td>78.7</td>
</tr>
<tr>
<td>romanian-swat_hk-bo</td>
<td>72.7</td>
<td>77.1</td>
</tr>
<tr>
<td>swat_hk-romanian</td>
<td>72.4</td>
<td>76.1</td>
</tr>
<tr>
<td>all_features MBL</td>
<td>71.2</td>
<td>76.4</td>
</tr>
<tr>
<td>Baseline</td>
<td>59.8</td>
<td>62.8</td>
</tr>
</tbody>
</table>

4 Discussion

Table 1.6 reports the most used features. For each feature the number of words that selected it is given (out of a total of 39 words). The average number of features used for disambiguation is 7.4 for nouns, 6.8 for adjectives, and 5 for verbs.

As reported before, the near context is a very good indicator for a word’s sense. The words surrounding the target word seem to be most helpful for disambiguation, but their relevance decreases as they get more distant.
Georgiana Dinu

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>romanian-swat_hk-bo</td>
<td>Supervised learning using Maximum Entropy with boosting, bag-of-words and n-grams around the target word as features</td>
</tr>
<tr>
<td>swat_hk-romanian</td>
<td>The swat-romanian and romanian-swat_hk-bo systems combined with majority voting. Swat-romanian combines three classifiers: cosine similarity clustering, decision list and Naïve Bayes, using bag of words and n-grams around the head word as features.</td>
</tr>
</tbody>
</table>

Table 1.5: Best-performing systems in SENSEVAL-3 Romanian task

<table>
<thead>
<tr>
<th>Feature</th>
<th>CT₀</th>
<th>NA</th>
<th>NB</th>
<th>CT₁</th>
<th>CT⁻¹</th>
<th>CP₀</th>
<th>CT₂</th>
<th>CP⁻¹</th>
<th>CT⁻²</th>
</tr>
</thead>
<tbody>
<tr>
<td># words</td>
<td>26</td>
<td>25</td>
<td>23</td>
<td>19</td>
<td>18</td>
<td>18</td>
<td>15</td>
<td>14</td>
<td>13</td>
</tr>
</tbody>
</table>

Table 1.6: MBL with per-word feature selection. Selected features from the target word. The nouns preceding and following the ambiguous word as well as its word form seem to play a very important role. The data is not large enough to permit the observation of patterns in the way nouns, verbs or adjectives are disambiguated. The words in the exercise seem to choose the features for disambiguation in a similar way, regardless of their part-of-speech. The only exception are adjectives which are biased towards choosing features extracted from preceding context (preceding noun, preceding tokens), unlike verbs or nouns, which prefer an extraction window centered around the target word. On average, a noun chooses 3 features from the left context and 3 from the right. For verbs, its on average 2 words on each side while an adjective chooses 3.4 features from the left context and 2.2 form the right one. This can be explained by the fact that in Romanian both predicative and attributive adjectives follow the constituents they modify, which are presumably good indicators for the sense of an ambiguous adjective.

One of the extreme examples of words that were disambiguated using a very small number of features is the verb căștiga (to win). By only using the word form of the verb and the word form of following noun, the disambiguation accuracy increases from the 52.2% baseline to 72.2%. Examining the training data gives clues to why this happens. The word has five senses but the predominant two senses are to gain money or some other material benefits, and to win a sports competition (or a contest, a trial). The noun after (NA) is a very good sense indicator in this case, as the object that the verb takes (which predominantly coincides with the first noun) is bani, dolari, mărci or lei (money or various currencies) for the first sense, and...
one of: partida, derby, meci (sport competitions) for the second sense. This leads to NA being the first feature that the algorithm selects, which on training data increases accuracy from 50.2% to 66.9%. Further on, the word form (CT₀) also helps distinguishing the two senses. For example the form caștigăm (third person, plural) is predominantly used within the winning a sport competition sense, as ‘our team (we) won the game’. In the second step of the algorithm CT₀ is thus the feature that brings the best improvement, increasing accuracy from 66.9% to 71.8%. Adding any of the other features in the next step of the algorithm results in accuracy drops varying between 0.5% and 12%, suggesting that for the word under consideration, any irrelevant information only adds noise to the data.

5 Conclusion

Results on a Romanian lexical sample WSD task confirm that memory-based learning techniques are well-suited for sense disambiguation. The highest performance is obtained by integrating a per-word feature selection algorithm, which allows each word to select it’s relevant, sense indicating, features. The system built this way outperforms all the other systems participating in the Romanian SENSEVAL-3 task even though it uses much less complex features.

Bibliography


of the 34th annual meeting on Association for Computational Linguistics, Morristown, NJ, USA, pp. 40–47. Association for Computational Linguistics.


Memory-Based Word Sense Disambiguation for Romanian with Automatic Feature Selection
An Incremental Model of Fragments in Dialogue

ANDREW GARGETT
King’s College London
andrew.gargett@kcl.ac.uk

Abstract. An approach is presented for modeling how context is (re)used in dialogue which leads to a fully incremental account of processing fragments. Examples of fragment clarification requests are outlined and classified, and previous literature on clarifications is discussed. Many previous accounts require pre-processing of the context. An account is then presented which models context in minimal terms as simply the structure of recent utterances. The resulting dialogue model is able to reuse context directly, without any form of pre-processing. Moreover, when taking their turn to speak, dialogue participants can start at any point, either from scratch or from what is recorded in context. The result is a fully incremental account of processing fragments in dialogue.

1 Introduction

Dialogue is replete with nonsentential utterances such as fragmentary forms of clarification requests (CRs).

Consider the following request for clarification of the subject of a statement (adapted from Purver 2006):

(1.1) A: Bill left.
   i. B: Bill (left)?
   ii. B: Bill?
   iii. B: “Bill”?
   A: Bill (left).

Case i. of B’s responses is a CR where a paraphrase involves A’s entire original utterance. There are three reasons for such a CR: (a) B understands A’s utterance, but is uncertain about the individual referred to, (b) confirmation of identity, (c) requesting information leading to identity. Cases ii. and iii. of B’s responses, on the other hand, are CRs with possible paraphrases restricted to the subject of A’s statement. These latter forms are

1This paper reports collaborative work being done with Ruth Kempson and Eleni Gregoromichelaki. I am grateful to them both for ideas and feedback.
arguably of three kinds: (a) Bill has been parsed, but B is uncertain and abandons the parse, (b) B fears she has misheard, and (having heard something) guesses what was said\(^2\), and (c) B explicitly asks for information to be repeated.

CRs have several interesting features. First, they repeat specific material from the context. Unlike standard questions, clarifications are not about requesting new information from interlocutors (as with WH-questions), and frequently involve repetition of items from (the immediate) context.\(^3\) Second, their brevity opens up a range of (possibly indistinguishable) interpretations. Third, they have a distinctive intonation (eg Rodriguez and Schlangen 2004). It is suggested here that this intonation is typically contrastive with the immediately preceding utterance being clarified (eg statement, question or request). The resulting clarification-response pair may be itself embedded within a larger adjacency pair (such as question/answer, statement/response, etc), suggesting a more general interactive repair mechanism in dialogue. This mechanism has been described as the basis for externalised forms of inference (Pickering & Garrod 2004). The present paper provides an example of a uniform grammar-internal characterisation of such mechanisms.

The claim here is that by applying the dialogue model of Purver et al. (2006), set within the Dynamic Syntax framework (DS, Kempson et al. 2001, Cann et al. 2005), interactions between CRs and fragment responses/replies (FR) can be modelled as incremental request/provision of clarification at arbitrary points in the dialogue. As we shall show, this requires neither lifting clarification fragments to yield some clausal type nor coercing the context to create a higher type suitable for combining with the fragment. Rather, context resolution occurs directly at the same level as the fragments themselves.

2 Previous Literature

As a form of nonsentential utterance (NSU), CRs have typically been modelled via pre-processing. Earlier approaches to NSUs have been either more syntactic, resolving them as structurally incomplete sentences (where missing information is assumed to be “hidden”), or more semantic, raising them to some higher sentential level.\(^4\) A third, more recent approach to NSUs (eg Ginzburg & Sag 2000, Purver 2004, Ginzburg & Cooper 2004, Fernandez-Rovira 2006), instead processes contextual information (which they term

\(^2\)For example, here B might say Bill and be right, or Jill and be wrong.

\(^3\)CRs may also be non-repetitive, for example, Jill’s husband? as a CR for Bill left, but I ignore these in this paper except to note that in principle mechanisms relating to apposition seem to be applicable to such cases, given the dynamic incremental approach to be advocated.

\(^4\)See various papers in Elugardo & Stainton 2005.
context coercion), whereby context combines with the propositional roles attributed to NSUs (eg CRs are ask moves, Purver 2004: 19). This latter approach involves a notion of incrementality, with phonological, syntactic, and semantic projection of sub-parts of complex signs being constructed in parallel, and as information becomes available.\(^5\) However, a stronger notion of incrementality is available to computational accounts, with structural projection following “word-by-word” processing as closely as possible, with direct access to the interaction between linguistic and contextual information. As Purver and Otsuka (2003) argue, this form of incrementality directly accords with psycholinguistic results. The DS dialogue model (Purver et al 2006) aims for this latter kind of incrementality.

3 Toward an Alternative Account of CRs

Dynamic Syntax (DS) is a parsing-based approach to linguistic modelling, with online processing and word-by-word update. The formalism consists of decorated binary branching trees representing predicate-argument structures, where interpretation involves goal-directed growth of tree decoration formalised using LOFT (Blackburn & Meyer-Viol 1994). Central to the account is the modelling of underspecification at all levels of tree relations, formula values, and tree-node identification, and update involves strictly monotonic information growth for any dimension of decoration. All aspects of underspecification have an associated requirement which are goal-directed elements driving update. For example, an underspecified subject node of a tree may have a requirement expressed in DS with the node decoration \(?Ty(e)\), for which the only legitimate updates are logical expressions of individual type (\(Ty(e)\)). This concept of requirement however is quite general, with types, formulae, tree relations, all enabling the formulation of corresponding requirements which must be satisfied in all successful derivations.

Amongst the most important of the concepts of underspecification is that involving a weakly specified tree-relation between a dominating node labelled \(Tn(a)\), and an unfixed node decorated with \(\langle \_\_\_\_ \rangle Tn(a), ?\exists x Tn(x)\). The requirement in this latter decoration guarantees that this relation will have to be updated to some fixed relation during the construction process. As this indicates, requirements are essential to the dynamics informing the DS account of CRs: all requirements must be satisfied if the construction process is to lead to a successful outcome.

Structure is built from such general computational actions and lexical

\(^5\)See the discussion of fractal heterogeneity in Ginzburg & Cooper 2004. Note further that this notion comes via Ginzburg and Sag (2000: 4), who suggest the apparent psycholinguistic plausibility of the incrementality of constraint-based approaches, citing among others Johnson and Lappin (1999). The latter explicitly refers to the notion of incremental correspondence (Johnson & Lappin 1999: 65-9), where this involves sub-parts of a linguistic item built “in tandem”.

77
actions. The role of computational actions is to dictate general processes for unfolding trees, introducing subject and predicate nodes, introducing and updating unfixed nodes, and, once individual nodes are successfully decorated (with no requirements), there are general processes for compiling interpretation for all non-terminal nodes in the tree, and a strict adherence for compositionality on the resulting tree. Lexical actions, on the other hand, are procedures for building structure from lexical items but expressed in exactly the same terms. Growth is invariably incremental and word-by-word, with overall structure being updated by the procedures associated with particular words as they are encountered. Importantly, the DS update algorithm is restricted to pointed partial trees, which is to say partial trees that have a pointer recording the parser’s progress. The pointer governs how parsing proceeds, and plays an interesting role in accounting for CRs.

Individual trees are thus taken to correspond to predicate-argument structures. To obtain more complex structures, a general tree adjunction operation is defined to license the construction of a tree sharing some argument term with the node from which the adjunction process is defined to apply, yielding so-called Linked trees (Kempson et al. 2001). Linking models the way that the nodes of separate trees can share a term, where some node in one tree links to the topmost node of another, depending on the relationship to be captured. Information from the resulting adjoined trees is modelled as a conjunction of terms at the node from which the link is made (if lower than the mother node, this may percolate upwards for the final version of the overall tree).

In any of the trees so induced, underspecification can extend well beyond the structural. An example of content underspecification is given by pronouns, whose intrinsic content constrains the value assigned the node being decorated. This is represented as a place-holding metavariable such as \( U \), as in the case of structural underspecification, with its associated requirement being a formula value \(?\exists x Fo(x)\). Names too, arguably, have a decoration of this form, the name itself serving as a constraint on the individual being picked out, to be represented by some individual constant. Thus the name Bill is assigned the linguistic content expressed by the decoration \( U_{\text{Bill}}(U), Ty(e) \), the subscript restricting possible updates to the metavariable. This subscript specification is shorthand for an instruction to induce a transition across a LINK relation to a tree whose topnode is decorated with a formula \( Bill'(U) \), the name being taken as a predicate on individuals.\(^6\) On any occasion of use then, names constitute a procedure for identifying the individual being talked about, a representation of whom has to be entered as decoration on the emergent tree.

In DS, generation is also goal-directed, and follows the parsing dynamics (this being the core mechanism). A speaker’s goal tree represents what

\(^6\)We suppress this building of a LINKed structure in all diagrams.
they wish to say, and each step licensed in the formalism constitutes the basis for some possible generation step. A major constraint on this is a subsumption relation between the constructed “parse” tree and the goal tree. Incremental (word-by-word) parsing, and lexicon search for words which provide appropriate tree-update relative to the goal tree, enables speakers to produce the associated natural language string. Since understanding involves parse trees, modelling dialogue in DS requires examining both goal and parse trees for speakers, and may also involve reconstructing the hearer’s parse tree.

Clarification is modelled by comparing such trees: the analysis seeks to show the extent to which B has successfully parsed what A has said, with the ability at any stage to interrupt to ask for clarification by producing either a repeat of the expression or some alternative. B’s parse tree thus records where miscommunication occurred. According to the general DS account of generation, such a repeat of the word is licensed only if B’s goal tree matches some parse tree that includes the relevant subpart of A’s utterance as an addition, this addition being what B is seeking to obtain clarification about. Thus, CRs characteristically involve a one-step transition from parse-tree to goal-tree which is in effect constituted by the actions being queried. Note that this amounts to a mechanism constraining the choice of goal tree: B chooses to repeat that part of A’s that is the source of the problem. Significantly, B can reuse the already constructed parse tree in their context, thereby starting at this point, rather than having to rebuild an entire tree. A key feature of the analysis here is that such reuse is a generally available option for dialogue participants.

The following analysis assumes a minimal dialogue structure:

(1.2) A: STATEMENT
       B: CLARIFICATION REQUEST
       A: RESPONSE

Following A’s statement, B makes a CR, to which A responds. Variation is possible here, as seen in example (1.1). Schema (1.2) involves three turns, where modelling each turn requires providing a goal and parse tree for the speaker, and a parse tree for the hearer. Three kinds of CRs will be considered here: one kind of non-constituent CR, and two kinds of constituent CRs.\footnote{Non-constituent CRs expressing surprise at the content of some utterance (that is otherwise completely understood) will not be considered here.} Figures 1.1 to 1.3 detail trees for a simple non-constituent clarification, where the utterance up to the predicate has been parsed, including the name \textit{Bill}, but it is unclear which person of that name is being referred to. The analysis will be extended to constituent kinds of CR in section (3.2).
Figure 1.1: Subject Non-constituent Clarification: Result of A’s statement

(a) A speaking; goal tree (left), parse tree (right)

Figure 1.2: Subject Non-constituent Clarification: Result of B’s CR

(b) B hearing
3.1 Non-constituent Clarifications

Consider Figure 1.1. A states that Bill left (Figure 1.1(a)), and B parses A’s statement. Note that B has parsed both subject and predicate, although requirements remain on the subject node. It is assumed that, in the initial parse of the subject node, B has not identified who is being talked about, so that node remains with a requirement for a formula value. Yet, monotonicity of tree growth is preserved in this derivation, despite the parse (hence production) involving return of the pointer to this node. Despite parsing the entire string, B remains unclear who is being talked about, and may at this juncture seek clarification. This is licensed by a variant of *Adjunction, called Late-*Adjunction (Cann et al. 2005), whereby a node decorated by \(?T_y(e)\) can be introduced, licensing the parse of the word Bill hence also its generation.

Figure 1.2 displays this step. Uncertainty about the value for Bill leads to B’s CR. B’s goal tree in Figure 1.2(a) is the same as their parse tree in Figure 1.1(b), since these trees effectively specify what actually went wrong for B. It is important to note that B need not start from a blank slate when taking their turn to speak; B is able, rather, to employ whatever is in their context recording the information from the most recent speaker’s utterance. Hence, B will start from their previous parse of A’s most recent utterance. Note that this is indeed what has been modelled in Figure 1.2(a), where B has used what is in their context (i.e., the results of parsing A’s most recent utterance).

---

\(^8\)Recall that in cases of nodes introduced as unfixed, the dominating node is labelled \(T_n(a)\), and the unfixed node, related to its dominating node only by a weakly specified relation is decorated with \(\langle \uparrow \ast \rangle T_n(a)\), indicated in the diagram with a dashed line.

---

Andrew Gargett

Figure 1.3: Non-constituent Clarification: Result of A’s response
utterance) as the basis for constructing the tree for their own utterance. Hence, in Figure 1.2(a), B’s parse tree has an unfixed node (signified by the dashed line) projecting from the subject node, where this is constructed via Late-*Adjunction. It is this move which provides a platform to enable B to repeat the word.\(^9\)

Now, CR can be modelled as an interactive strategy for repairing misalignment (eg Pickering & Garrod 2004). Recall that the pointer governs the progress of parsing. For example, in Figure 1.1(b), the pointer remains on the subject node, signifying B’s being unable to specify $Fo(U Bill')$. However, A has a complete parse tree, that resulting from A’s own utterance, with the pointer residing at the topmost node. Yet, interactive repair requires A and B to start at the same node, which is to say, the subject node. So how does B signal to A where to start? This is where B’s intonation for the CR plays a crucial role, indicating that the pointer in A’s parse tree is to move to the node decorated by the isolated expression. Now A can parse B’s utterance by the projection of an unfixed node (Figure 1.2(b)), achieving sufficient alignment with B for successful communication. Notice how this promotes alignment between the representations of the interaction available to A and B, without either having to directly model each other’s mental states, or indeed without having to process any other information outside of that recorded in their parse of the most recent utterance.

The parse tree can also be updated via another kind of action: the pragmatic action of Substitution. This replaces metavariables still residing on trees with contextually provided terms. In particular, this enables specifying the information provided by anaphoric expressions, where this always requires update via context (Purver et al. 2006). Substitution provides the correct value for $Bill$ from context, which here involves substitution by $m_{21}Bill'(m_{21})$, this making it possible to complete both A’s and B’s tree structures (Figure 1.3). Note that in this account re-alignment is driven by underspecification and the means available to the participants to resolve it in a thoroughly incremental way through reusing context. Rather than forcing revision or contextual coercion of any kind, the CR simply throws up a form of underspecification, which it is then up to the respondent to resolve.\(^{10}\)

### 3.2 Constituent Clarifications

Constituent clarifications are essentially similar to non-constituent ones, the only difference being the stage in the construal process which the hearer B is able to reach. If B can reach full understanding, she may nevertheless ask

\(^9\)Of course, B could choose to use some other term, provided that is also an anaphoric expression able to be identified as the term denoted by the name itself, which may account for non-repetitive clarificatory fragments.

\(^{10}\)Of course, a non-cooperative partner may well choose not to.
for clarification and this will constitute a so-called clausal reading. If B can parse the word and identify who is being talked about, she may choose to query this immediately, before proceeding to process the remainder. If B is able to parse the words, but not assign a suitable referent-denoting term (in the case of names), then she may nevertheless be able to provide some tentative decoration - the place-holding strategy provided as the intrinsic content of the name. Indeed it is notable that unless she signals total failure to understand part or all of what is said, for example, by saying Who?/What?, B has to make some judgement as to which name is used. Moreover, since every word is associated, by definition, with actions for tree-growth, any word that is presented as a CR will simultaneously present a putative tree-update, hence be characterisable in essentially the same terms as so-called clausal readings. Both involve some putative update to the tree about which the hearer is requesting confirmation. The CR strategies involved are basically the same as for the non-constituent cases. Of particular interest here are cases where: (1) Bill has been parsed, but uncertainty leads B to abandon the parse at this point, and (2) B mishears, and guesses what was said (eg guessing correctly with Bill, or incorrectly with Jill). For case (1), the initial trees are the same as Figure 1.1, except for the important difference that a requirement remains on the predicate node of B’s parse tree (ie B stopped parsing A’s statement immediately after Bill). Thus, CRs involving repetition are a strategy enabling A to reuse contextual material, rather than having to build any new structure.

For case (2), B mishears the initial consonant segment /b/, guessing the word as /bill/. This guess could have been /jill/ where /j/ represents the voiced palatal plosive, but it is here taken to be /bill/ for ease of analysis. In contrast to Figure 1.1(b), although the pointer rests at the subject node, both subject and predicate nodes remain underspecified, where B is unable to retrieve lexical information. Through guessing, the subject node of B’s parse tree is completed. Thus, B supplies a value for the denotation of Bill, where this value could be fixed as a contextually provided term such as \( m_{34} \text{Bill'}(m_{34}) \), or else as the still underspecified \( \text{U}_{\text{Bill}}(\text{U}) \) (the latter reflecting uncertainty as to the word itself). Note that once again, A’s parse tree will reflect the usual parsing strategies for names, with an unfixed node projecting from the subject node as a consequence of application of Late-*Adjunction.

An important feature of the DS analysis is that it enables the distinction between constituent and non-constituent CRs to be downplayed. However, given the generality of the tree-growth mechanisms as applied strictly incre-

---

11 One interesting difference in this example is that the actual guessing itself might involve a step of abduction.

12 Although note that non-repetitive CRs remain in need of a full account. And no account is given here of what is involved in explicit representation of interlocutor’s roles in the utterance with questions such as “Why did you use that word?”
mentally at any word or phrasal level, there is reason to remain confident that the various types of clausal and constituent readings can be seen to fall under the general pattern.

4 Discussion

The approach taken here avoids coercion of either the utterance or the context, the advantage in terms of any processing algorithm being that at least these operations are not required. Rather, all that is required is a means of directly reusing available contextual material. This is particularly the case where repetition is involved. Yet, even in cases of CRs lacking repetition, there is a way of processing these in much the same way as has been done for examples containing repeated material: what is required is a more general means of representing how such anaphoric expressions can be identified as the term denoted by the name, and as suggested above, DS currently provides the basic machinery for such an account. Possible extensions of the current approach for cases where B has to (perhaps abductively) make a guess at what has been said (eg despite some perceptual error such as mishearing the initial consonant) have also been illustrated.

Any algorithm constructed for handling CRs in DS will also need to be extended to a range of other cases, such as clarifications of questions (see Purver 2004 for taxonomy of clarification phenomena). Given the relative simplicity of this initial model of CRs, it should be straightforward to extend this to these other cases. It is interesting to note that the various features specific to CRs, such as repetition and contrastive intonation, are readily incorporated into a parsing account. Further, no additional (pre-)processing of either utterances or contexts is needed, as everything that is required is directly available for immediate employment by interlocutors. This accords with Pickering and Garrod’s (2004) challenge to model dialogue in its own terms (rather than eg as a special case of monologue). The bonus of this approach is its intrinsic incrementality: the underspecification that emerges through interaction needed for clarification (and repair) is resolved directly via the dynamics of parsing.

5 Bibliography


13For more on the role of intonation in the parsing-based account of DS, see Kiaer 2007.
Andrew Gargett


An Incremental Model of Fragments in Dialogue
Foundations of Semantic Role Annotation:  
An Entailment-based Annotation Scheme

VOULA GOTSIOULIA  
University of Essex  
pghots@essex.ac.uk

ABSTRACT. This paper presents a lexical entailment view to semantic role annotation. Lexical entailments, i.e. semantically well-founded, clearcut, generic notions are assigned to semantic arguments of predicates. I discuss the potential of applying an entailment-based annotation layer for the acquisition of systematic knowledge about linking to syntactic form.

1 Introduction

The creation of large-scale lexical semantic resources that provide relational information about lexical items, in particular information about predicate-argument structure, is at the heart of current NLP research. Corpora with semantic role annotation (markup) form the basis for training and evaluation of semantic parsing algorithms that automatically identify the semantic roles conveyed by sentential constituents. In addition to the development of models for automated semantic role assignment, such corpora furnish the essential data for the acquisition of linguistic knowledge at a general syntax-semantics interface. Systematic mappings of argument structure to syntactic form can be formalised and integrated into alternative NLP systems (Frank, 2004). Furthermore, generalised role assignment patterns can be applied to semantic parsing models to remedy the data sparseness problem.

In this paper I address primary design issues of semantic role annotation. I discuss the implications of the roles used for the markup in the above application context. I consider Dowty’s (1991) influential work on semantic roles, the proto-role theory, as well as Wechsler’s (1995) and Davis’ (2001) refinements to it. Relying on their insights, I propose an annotation scheme in which well-founded, clearcut, generic notions, i.e. lexical entailments, are assigned to semantic arguments of predicates. I discuss the effect of applying a lexical entailment annotation layer to extract generalisations about the syntactic realisation of semantic roles.
Foundations of Semantic Role Annotation: An Entailment-based Annotation Scheme

2 Semantic Role Annotation and Corpora

Two corpora are available for English representing different approaches to the prickly notion of semantic role: PropBank and FrameNet.

The Proposition Bank (PropBank) (Kingsbury and Palmer, 2002) is a one million word corpus in which predicate-argument relations are marked for every occurrence of every verb in the Wall Street Journal (WSJ) part of the Penn TreeBank (Marcus et al., 1994). PropBank assumes a combination of syntactic and semantic cues for creating annotations. Different senses of predicates are distinguished mostly on syntactic grounds. For each sense, arguments are numbered sequentially and tagged with labels from ARG0 up to ARG5. Argument labels are arbitrarily defined on a per-verb basis. However, they are used consistently across syntactic alternations for the same verb meaning. Adjuncts are marked with the tag ARGM and one of a set of ‘functional tags’ denoting the role of the element within the predicate, e.g. ARGM-LOC for locatives, ARGM-TMP for temporals. Example PropBank annotations:

(1) \[[\text{ARG0 Blue-chip consumer stocks}] \text{[rel provided]} [\text{ARG1 a lift}] \text{to} [\text{ARG2-to the industrial average}].\]

(2) In addition, \[[\text{ARG0 the bank}] \text{has an option * to} \text{[rel buy]} [\text{ARG1 a 30 \% stake in BIP}] \text{from} [\text{ARG2-FROM Societe Generale}] \text{[ARGM-TMP after Jan. 1, 1990]} \text{at} [\text{ARG3-AT 1,015 francs a share}].\]

PropBank does not attempt to formalise the semantics of the roles it defines. Argument labels have no theoretical significance in that no consistent mapping is ensured between a label and a semantic role. This is particularly clear with higher-numbered labels. For instance, ARG2 may indicate ‘Beneficiary’ for one verb, while for another it may indicate ‘Source’, as in (1)-(2).

Lower-numbered labels map to various roles as well, though they are generally consistent across verbs. ARG0 and ARG1 are defined syntactically as well as semantically: ARG0 is assigned to subjects of transitive verbs and corresponds to traditional Agents, Experiencers, certain types of Theme, etc.; ARG1 is assigned to objects of transitive verbs and subjects of certain intransitives and is the equivalent of Patient, Theme, Stimulus, etc. PropBank labels do not lend themselves to any abstraction of syntactic-semantic information. PropBank itself refers to verb classes defined by VerbNet\(^1\) mapping the labels to semantically coherent roles to ensure their consistency for verbs within the same class.

---

\(^1\)VerbNet (Kipper et al., 2000) is an implementation and refinement of Levin’s (1993) verb classes. These are defined on the basis of the ability of verbs to participate or not in pairs of syntactic frames representing alternations in the expressions of arguments (diathesis alterations).
While PropBank implements a corpus-based approach, FrameNet\(^2\) is primarily a semantic lexicon based on Fillmore’s (1985) theory of frame semantics. In FrameNet, word meaning is represented by frames, i.e. schematic representations of stereotyped situations encoding certain amount of real-world knowledge. Each frame is associated with a set of lexical items (verbs, nouns, or adjectives) that evoke it and a set of roles (frame elements) corresponding to the participant roles in the situation. A distinction is made between core and non-core (marginal) roles.

FrameNet includes manually annotated examples from the British National Corpus and provides a layer of grammatical function annotation (Fillmore et al., 2003). Currently it contains more than 625 frames covering more than 8,900 lexical items. The following sentences exemplify the supply frame in which “a supplier gives a theme to a recipient to fulfill a need or purpose of the recipient”:

(3) She sat down on her bed, carefully folding her clothing and packing it into [theme the small carryall] [supplier Starfleet] had provided [recipient her] [purpose_of_recipient for the journey].

(4) [supplier I] have equipped [recipient my leechtroopers] [theme with tiny subspace displacement thingummies].

FrameNet avoids the pitfalls of attempting to define a small set of semantic roles by defining roles in terms of frames. Frames are situated in semantic space by means of a number of directed (asymmetric) lexical semantic relations that generally hold between a Superframe (a less dependent, or more abstract frame) and a Subframe (a more dependent, or less abstract one)\(^3\). A theory of frame-to-frame relations and frame element identities or analogs across frames is required to enable the formulation of generalisations concerning mapping to syntactic form. Nonetheless, such a frame-wise approach misses regularities across completely unrelated frames.

### 3 Lexical Entailments and Argument Structure

In this section I consider an alternative approach to semantic roles. From a strictly theoretical point of view, Dowty (1986) has argued that role types like Agent, Patient, Theme, etc. are ill-founded inasmuch as it is difficult to establish sets of properties that pick out unified notions. Dowty (1991) refrains from the idea of semantic roles as discrete categories. He describes argument selection (i.e. the question of what principles determine which argument of an n-place relation denoted by a predicate is expressed by which grammatical relation) in terms of prototypical properties entailed by verbal

---

\(^2\)http://framenet.icsi.berkeley.edu

\(^3\)A detailed description of these relations can be found in the FrameNet Book (Ruppenhofer et al., pp. 104-111).
Foundations of Semantic Role Annotation: An Entailment-based Annotation Scheme

Dowty employs two concepts, that he calls Proto-Agent and Proto-Patient, as suitable abstractions for the following lists of entailments:

(5) **Contributing properties for the Agent Proto-Role:**
   a. volitional involvement in the event or state
   b. sentience (and/or perception)
   c. causing an event or change of state in another participant
   d. movement (relative to the position of another participant)
   e. (exists independently of the event named by the verb)

(6) **Contributing properties for the Patient Proto-Role:**
   a. undergoes change of state
   b. incremental theme
   c. causally affected by another participant
   d. stationary relative to movement of another participant
   e. (does not exist independently of the event, or not at all)

No unifying semantics underlies each of the lists in (5) and (6). Proto-Agent and Proto-Patient are ‘higher-order generalisations about meanings’, so that it misrepresents Dowty’s position to speak of a particular argument of a predicator as the Proto-Agent or Proto-Patient. The semantics of predicates involve combinations of entailments that map directly onto syntax according to the following numerical procedure:

(7) **Argument selection principle:**
   In predicates with grammatical subject and object, the argument for which the predicate entails the greatest number of Proto-Agent properties will be lexicalised as the subject of the predicate; the argument having the greatest number of Proto-Patient entailments will be lexicalised as the direct object. (Dowty, 1991: 576)

In a related vein, Wechsler (1991, 1995) considers argument structure and linking to syntactic form in terms of universal semantic primitives, i.e. concepts that are independently motivated and required by the semantics of natural language. One such primitive is NOTION that reconstructs Dowty’s sentience entailment by assuming a 2-place ‘notion’ semantic relation\(^4\). For transitive verbs entailing NOTION a generalised linking pattern suggests that the individual expressed by the subject is entailed to have a notion of the individual expressed by the object, while the reverse entailment pattern does not necessarily occur.

\(^4\)It is defined technically in situation semantics but corresponds fairly straightforwardly to its intuitive sense.
A primitive PART relation applies a mereological theory of event structure to predicates denoting causal events with causally affected participants. It identifies roles for which a change of state in the participants that fill them reflects the temporal structure (i.e. progression) of the denoted event. Such roles (**incremental themes** in the terminology of Dowty) tend to be objects of transitive verbs or subjects of intransitives, but not subjects of transitives.

(9) The acid dissolved the metal.

In a different context the primitive PART relation accounts for argument selection of stative verbs that do not involve any notions or causally affected objects. These include the so-called ‘container’ verbs. An emerging generalisation is that the object participant is part of the subject participant, while the reverse pattern does not go through.

(10) The book includes an appendix.

While Dowty’s proto-role theory is restricted to the domain of monotransitive verbs (the ones with a subject and a direct object), Wechsler considers verbs with prepositional complements as well. He argues in favour of the view that many prepositions heading complement PPs are semantically contentful rather than simply tagging a complement of the verb. Furthermore, their semantics must unify with that of the predicator. This view of prepositional complements has been developed by Gawron (1986), who points out that prepositions that occur felicitously as complements of a given verb are not random.

(11) a. long for, yearn for, wish for, hope for...

    b. hanker after, thirst for/after, aspire to/towards...

Gawron posits that a necessary (though not sufficient) condition for a preposition to be selected for a given complement of a verb is that the prepositional semantics be compatible with (or ‘a component of’) the semantics of the verb. Some degree of arbitrariness is imposed by individual lexical stipulations, as in (11b).

---

5In (8)-(10) I reproduce some of Wechsler’s examples.

6Wechsler notes that for-PPs also occur as adjuncts with the same desiderative sense: *John ran for cover when it started to rain.*
4 Annotating the Entailments

Semantic role entailments establish a generic, semantically well-founded way of describing argument structure and semantic roles. Contrary to cluster labels like traditional semantic roles, role entailments are fine-grained, clearly defined, and thus straightforward to track down. In what follows I propose an annotation scheme in which semantic arguments of predicates are tagged with lexical entailments. I assume a tentative set of properties that account for a broad range of predicators displaying various syntactic patterns beyond transitivity. I strongly rely on Davis’ (2001) work that extends Dowty’s and Wechsler’s semantic notions. Some further clarifications and example annotations follow.

**Notion** applies Wechsler’s ‘notion’ relation. Semantic role entailments are not intended to classify arguments in a one-to-one fashion. For instance, in situation types in which a participant is entailed to have a notion of more than one entities the entailment ‘conceived/perceived’ might be assigned to more than one arguments, as in (13). Similarly, each argument should be annotated with all entailments furnished by the verb meaning. In effect, the semantics of predicators involve various combinations of entailments.

(12) [conceiver I] noticed [conceived their appearance] and also noticed [conceived that, left alone, they disappeared too].
(13) [conceiver She] accused [conceived him] of [conceived not trying enough to save them].

**Causation** involves a causer and a causee that is physically or mentally affected. Change-of-state and incremental themehood are additional entailments that may hold of the causally affected participant. A combination of causation and notion is exemplified in (15). That is, the children are entailed to be causally affected and have a notion of the stimulus that affects them. The stimulus itself (the movie) is both a causer and a perceived entity. For the readability of the examples only one entailment is displayed at a time.

(14) [causer The wind] destroyed [causee, ch-of-st the roof].
(15) [causer The movie] frightened [causee the children].

**Volitional involvement** is one of Dowty’s proto-role entailments. It refers to participants characterised by conscious choice, decision, or intention. The verb *murder* is a typical example of volitional action, as one cannot murder accidentally (contrary to *kill*). However, the degree to which such a property may be considered to be intrinsically tied to the semantics of predicates is highly debatable; that is, volition is often implied by the entailments that are necessary to ensure that a predicate’s meaning necessarily involves a particular entailment in all possible environments.

---

7 Generally, certain diagnostics should be applied to ensure that a predicate’s meaning necessarily involves a particular entailment in all possible environments.
context rather than the predicate’s core meaning, as is illustrated in (17a-b).

(16) [VOLITIONAL Sam] murdered the dealer in his house.
(17) a. [VOLITIONAL The police] chased the thief.
   b. One of the remaining young fish was being chased around the tank by [?VOLITIONAL an older Harlequin], which had been in the aquarium for a few years.

Verbs like *murder* or *chase* should be annotated in terms of notion rather than volition. I additionally propose to replace volition with the entailment ‘control of action’ assigned to participants that control the course of action denoted by the verb, i.e. its start, intermission, or end point. This entailment is necessary for situation types where some action takes place and no notions or causally affected participants are necessarily involved. It distinguishes, for instance, verbs of self-motion from verbs of other type of motion that are described next.

(18) [CONTROL The horse] kept running in the field.

**Motion** involves a moving entity and a stationary reference frame (*path*) within which various points (start, end, or intermediate) may be further specified. Caused motion verbs like *run* in (21) involve a participant that both moves and is causally affected (i.e. set to motion).

(19) [MOVING I] entered [PATH the room].
(20) [MOVING A boat] drifted loose on [PATH the water].
(21) John ran [MOVING the car] into [PATH the field].

**Impingement** or **forceful contact** is added to the list of entailments by Davis. It refers to situations denoted by verbs like *hit*, *strike*, *poke*, *press*, *push*, *pull*, etc., where one participant is entailed to exert force on another while the impinged-on participant is not entailed to be affected in any way.

(22) [IMPINGER He] hit [IMPINGED-ON the ground].
(23) [IMPINGER Mary] pushed [IMPINGED-ON the door] to open.

**Inclusion** applies Wechsler’s primitive ‘part’ relation relying on the notion of relative size. A similar asymmetric relation between two entities or quantities compared to each other is identified by the entailment **surpassing**.

(24) [WHOLE The game] includes [PART dice].
(25) In her time [SUPERIOR she] outstripped [INFERIOR most politicians] in the country.
Possession is introduced by Davis to account for transitive verbs like have, own, acquire, inherit, lack and also ditransitives like give, send, etc. The latter have been analysed as meaning ‘cause someone to possess something’, i.e. in terms of causation and possession. The eventual possessor is equivalent to the traditional role ‘recipient’.  

(26) [POSSESSOR Iran] had acquired [POSSESSED four nuclear weapons] from former Soviet Moslem republics.
(27) [POSSESSOR He] finally obtained [POSSESSED his dream house].
(28) Daniel gave [POSSESSOR Mary] [POSSESSED flowers].

Possession applies to verbs of commercial exchange as well. These verbs involve two possession transfer events, the transfer of goods and the transfer of money, either of each might be ‘highlighted’.

(29) Ben sold [POSSESSOR Lisa] [POSSESSED the car] for 3000 euros.
(30) Lisa paid [POSSESSOR Ben] [POSSESSED 3000 euros] for the car.

Prepositional complements are marked with the preposition and a tentative meaning conveyed by it. Following the essentials of Gawron’s account, I assume that prepositional phrases filling necessary or optional slots of verbal semantics are compatible with the corresponding verbal entailments to which they may contribute additional information.

(31) [CONCEIVER He] looked for [FOR_DESIRE an empty space].
(32) [CONCEIVER Some] may indeed dream of [OF_CONCEIVE a united Germany].
(33) [CONCEIVER Ann] complained about [ABOUT_TOPIC the threatening call].
(34) [CONCEIVER The journalist] reported on [ON_TOPIC the US policy].
(35) [CONCEIVER Very few] trusted in [IN_CONCEIVE a future of change].
(36) John sold [POSSESSED his books] to [TO_POSSESSOR the second-hand bookshop].
(37) He already talked to [TO_CONCEIVER me] about [ABOUT_CONCEIVE this stuff].
(38) John ran [CAUSEE,MOTION the car] into [INTO_PATH the field].

*In fact, some ditransitives of this type involve an intended recipient rather than an actual one. This is particularly clear with verbs such as send and mail. We could thus additionally specify the weaker entailments ‘intended possessor’ and ‘intended possessed’.
5 Entailments and Linking Generalisations

Currently available resources such as VerbNet and FrameNet furnish linking information in terms of semantically related predicates. VerbNet espouses a strong version of a semantic basis of linking by defining classes of verbs that share the same syntactic alternations. A weaker view underlies FrameNet; that is, lexical units evoking the same frame are expected to have certain combinatorial properties in common. A related assumption is that if the semantics of one verb is a more specific instance of the semantics of another, then the linking of arguments in the semantically more specific verb represents a possible linking pattern for the semantically more general verb. Thus specifying generalisations of syntactic mappings across verb classes/frames essentially relies on a (multiple) inheritance semantic hierarchy, which in turn requires establishing correspondences between the specific roles assumed within each class/frame.

A significant asset of an entailment-based approach, on the other hand, is that argument structure is associated with surface realisations in terms of a limited set of abstract semantic notions. Thus generalised linking patterns can be extracted directly from data representing various types of predicates. Furthermore, they can be applied to novel data without reference to lexical semantic classifications and hierarchies. Such patterns include lexicalisations of individual entailments (like the ones discussed in section 3), bundles of overlapping entailments, and entailment-preposition combinations useful for abstracting over role-preposition correspondences. In (39) we note some generalised occurrences of particular prepositions with verbs denoting cognition (i.e. notion).

(39)  

a. insist on, bet on, count on, bank on, report on, comment on, concentrate on, meditate on, ruminate on, sleep on

b. reminiscent of, complain of, boast of, expect of, judge of, accuse of, speak of, imagine of

c. believe in, trust in, interest in

The prepositions above are associated with frame-elements local to distinct, possibly unrelated frames in FrameNet. In a frame-wise approach linking generalisations may emerge by analysing the distribution of role assignments for a given frame, abstracting over the specific lexical items and attempting to unify the abstractions across the lexical semantics hierarchy. A generic, entailment-based apparatus for translating frame element information and applying this translation to a distinct annotation layer is useful for methodologically decoupling linking information from specific lexical semantic content. Generalisations such as (39) can be formalised into general classes of non-lexicalised frames specifying syntactic mapping constraints that apply across frames.
6 Conclusions

I have presented a lexical entailment approach to semantic role annotation. Lexical entailments pin down argument structure relations in terms of a unified domain of linguistic data, i.e. the argument selection problem. They are firmly grounded in semantic intuition and have a wide coverage over relations that humans express in a systematic way.

Some classes of predicates that I have not discussed here raise interesting questions for an entailment-based approach. Verbs of location have not been considered at all. Many of them are so-called ‘symmetric predicates’ displaying great variability in linking patterns. Symmetric predicates generally involve arguments that are indistinguishable in terms of entailments. Furthermore, none of the previously mentioned entailments seems to hold of the semantics of verbs like resemble, match, involve, fit, suit, precede, follow, rely, and many others. The list of entailments that I have considered thus far is by no means complete. Finally, a different issue is to do with the metaphorical/contextual usages of predicates where the core entailments may be arbitrarily ‘relaxed’.

Future steps for pursuing an entailment-based approach and addressing its feasibility include (i) annotation of a representative data set from FrameNet and design of an architecture to extract linking patterns, and (ii) refinement and extension of the list of entailments.

Acknowledgements

My thanks are due to the Greek State Scholarships Foundation for granting my Ph.D. studies at the University of Essex. I am grateful to Stella Markantonatou, Massimo Poesio, and my supervisor Doug Arnold for their constant guidance and support. Special thanks go to the anonymous ESSLLI Student Session reviewers for their insightful comments and suggestions.

Bibliography


Aspectual Shift via Supervaluation

SABINE GRÜNDER
Department of Computer Science and Department of Linguistics, University of Leipzig
gruender@uni-leipzig.de

Abstract. Aspectual shifts in connection with temporal adverbial modification usually are treated by close analogy with type-coercion in programming languages. The paper shows that by taking a good metaphor too literally, semantic research so far did not do full justice to the kind of flexibility observable in the natural language examples. A new proposal is developed within the framework of finite-state temporal semantics combined with a supervaluational concept of underspecification. The simple shifting algorithm used in the present approach generates the correct set of possible readings on the basis of linguistic input only and, furthermore, may claim cognitive plausibility.

1 Introduction

There is a long tradition in philosophy of language and linguistics that has attempted to explain the way sentences express the temporal structure with respect to which they are to be interpreted. From the perspective of research in the field of semantics and syntax of natural language, differences in situational structure have been shown to have influence on, for instance, truth conditions and entailments, temporal sequencing in discourse, grammaticality of imperative, progressive, pseudo cleft and adverbial constructions. These observations helped isolate aspectually relevant properties, and gave rise to a number of tests which are now commonly used to distinguish at least four classes of situations: states, activities, accomplishments and achievements.

While, sometimes, aspectual classes are thought to be a means to classify verbal lexical entries, a closer look at the data shows that the aspectual class of a verbal phrase can change during the process of semantic composition under the influence of nominal arguments, prepositional phrases, tense and aspectual markers, aspectual auxiliary verbs and temporal adverbials. The aim of a theory of aspect, is therefore to predict the temporal structure of the situation denoted by a sentence on the basis of semantic information associated with certain of its syntactic constituents.

In the last twenty years, considerable progress has been made in describing the aspectual contribution of nominal and prepositional arguments (e.g.
But as far as the aspectual behaviour of temporal adverbials is concerned, some puzzling set of data still awaits proper explanation and appropriate formalization. From the very beginning of linguistic research on aspect, compatibility with “for”- and “in”-adverbials figured most prominently under the tests available to tell apart telic from atelic situations. The examples under (1) and (2) shortly repeat the well-known pattern.

\[(1) \text{be quiet for an hour / *in an hour}\]
\[\text{swim for an hour / *in an hour}\]
\[(2) \text{walk a mile *for an hour / in an hour}\]
\[\text{arrive *for an hour / in an hour}\]

Accordingly, atelic states and activities admit modification by a “for”-adverbial, but give rise to a marked interpretation when combined with an “in”-adverbial; for telic accomplishments and achievements things are just the other way around.

However, those claimed test cases do not seem to be as simple and clear cut as one would like to have them since, rather than being fully ungrammatical, the marked sentences display some kind of derived meaning. The focus of the temporal adverbial under (1) can be understood as being shifted from the situation itself towards its pre-phase, which is to culminate in the swimming process or the state of being quiet only an hour later. In case an implicit boundary for the swimming activity is being given by context (imagine a triathlon competition, for instance,) the process itself too can be modified by the adverbial after all. An iterative interpretation appears to be derivable for the examples under (2), with the “for”-adverbial giving the duration of the whole complex situation. Another possibility here is to let the adverbial fix the duration of the preparatory phase, i.e. the change ment’s being under development, preceding the culmination implicit in the underlying verbal description. Sometimes, a “for”-adverbial combined with a telic situation also seems to be able to modify the result state following its culmination, rather than any kind of process, as shown in (3).

\[(3) \text{leave the room *for an hour / in an hour}\]

These phenomena\(^1\) in connection with the application of temporal adverbials to situations that are - under the perspective of traditional aspectual theory - not suitable for them have proven to be notoriously difficult to treat in every kind of available framework. At the same time, the observed cases do not have the status of real exceptions, but appear crosslinguistically and spread over the whole verbal lexicon. Therefore, no theory of aspect can be seen as fully adequate as long as not offering a conclusive, systematic

\(^1\)Here and in all that follows I disregard the possibility of generic interpretations, as these seem to obey semantic principles different from what will be used to handle the group of examples presented above.
description and plausible explanation for this flexibility in adverbial semantics. The aim of this paper is to work out a formally simple analysis of the cases in question which should not only be able to generate the correct set of possible interpretations in a fully compositional manner, but also explains the underlying semantic mechanism in terms of a more general cognitive principle.

2 The State of Art: Type-Coercion

Before turning to my own proposal for an analysis of meaning shifts in connection with temporal modification, I will briefly discuss the idea figuring prominently in most of the recent theories (e.g. [2] [9] [16] [17] [18] [19] [22]). Moens and Steedman [11] conceived temporal adverbials as “(...) functions which ‘coerce’ their inputs to the appropriate type, by a loose analogy with type-coercion in programming languages”. Under the perspective implicit in the quotation, aspectual shift is triggered in the cases under consideration by a conflict between the aspectual type of the situation to be modified and the aspectual constraint set by the temporal preposition heading the modifier. Coercion operators, then, are thought to adapt the verbal input by mapping one sort of situation onto another. The underlying model is commonly supposed to include situations as first-order objects of the four basic sorts previously mentioned, i.e. states, activities, accomplishments, achievements (with the difference between the last two sometimes being neglected).

A first problem comes up immediately from this constellation. Since the temporal prepositions constrain the type of the situation only as far as telicity is concerned, the relation established by coercion cannot be functional, as it leaves us with several different possible outputs. And still worse, when looking at the given examples, it becomes clear that even when confined to one of the primitive ontological sorts, the output value of the operation remains underspecified, because after applying a “for”-adverbial to an accomplishment, for instance, the iterative and the result state interpretation alike count as stative. Since, on the other hand, in the case of real type coercion in programming languages the introduced relation has always to be functional, a first clear difference between the shifting operations in the two domains must be noted. Simply mapping one primitive semantic sort onto another cannot be what is formally going on in case of meaning shifts triggered by temporal adverbials, as this leaves us with no explanation for there being this exact number and kinds of readings one actually finds.

There are examples (i.e. "hiccup all day") where aspectual shifts appear to follow solely from conceptual background knowledge concerning default durations of certain sorts of events, rather than being triggered by a conflict in aspectual type requirements. I take these phenomena as laying outside the scope of any formal semantic treatment, and will not deal with them any further here.

2
Moreover, when taking a closer look at the explanations commonly given in the debate, one finds that wrapped inside the so-called type-coercion function there are in fact relations of a very different kind, establishing links between the atomic elements in the model such that these get related in a conceptually much more fine-grained way than a simple mapping between two logical types could have induced. Examples of the relations commonly used are Result, Iteration, Preparatory Phase. Accordingly, although the whole process appears to be triggered by a type conflict at the very first, these connections, established inside some kind of ontological network, actually come into play as a second, seemingly independent factor. They are actively conducting the shifting operation in its various directions by following some underlying principle which remains entirely unknown. Similarly, theories of aspectual coercion remain silent or confine themselves to vague, intuitive circumscriptions concerning the decisive question of how these relations are precisely to be defined\(^3\). Actually, from the point of view of the type-coercion paradigm in programming theory, these problems come up as a surprise, as this implicit, intermediate step on the way to the resolution of sort conflicts does obviously not have any counterpart in pure type coercion algorithms.

Consequently, the analogy between aspectual shifts triggered by temporal adverbials and type coercion as used in the programming language domain is, I claim, not as close as generally assumed. Applying the formal mechanism used in the latter case too strictly and blindly to the former, inevitably leads to the two interrelated problems mentioned above, and seems by now to obstruct a deeper understanding of what is actually going on when senses start floating like driven by themselves.

3 The Framework: Situations as Strings

In this section, I will (very) shortly introduce a decompositional approach to event semantics within the framework of Dynamic Logic \([7] [8] [13]\) that has been proposed by Naumann \([14] [15]\), together with a finite-state version of it developed by Fernando \([3] [4] [5] [6]\). What made me studying and, consequently, applying this kind of formalism was the intuition that what is needed to appropriately model aspectual shifts is the concept of a situation as a complex but logically coherent, dynamic unit. The more general theoretical decision lying in the background of the analysis taken up here is the move from traditional model theoretic semantics towards cognitive semantics, with a switch in theoretical perspective that takes aspectual cat-

\(^3\)This, I would claim, even holds true of Pulman’s proposal \([17]\), because the inherent connection in question gets out of sight of the theory by disappearing behind the interpretation function “I”. Beside this, the theory cannot account for all possible types of shifts; consider, for instance, the pre-phase interpretation of example (1).
egories as “(...) ways of viewing a happening, rather than intrinsic properties (...) of objective reality and the external world”[20]. This “representational
turn” now makes the semanticist describing descriptions of structures rather
than structures themselves and, by doing so, opens up the possibility of an
improved treatment of ambiguities in terms of 
underspecification [13]. The essential idea of this paper can be seen as being built on that spirit.

Quantificational Dynamic Logic, as known from program verification, offers
a convenient formal tool for decomposing situational concepts by looking
at them as some kind of state transformers. According to Naumann, verbal
lexical information consists of a program gradually changing the value of
a variable, which is normally corresponding to a particular property of the
verb’s object (incremental theme) or has been contributed by co-occurring
prepositional phrases. The quantificational information coming from the
determiner or preposition, respectively, controls the execution of a while-loop
via a boolean condition and either makes the program terminating after a
definite number of steps, or lets it going on indefinitely. The following example
gives the translation within the framework of the telic phrase “eat three
apples” (the domain of apples is structured as a complete partial order, with
⊕ being an operator on chains in this order such that x ⊕ 1 gives the least
of all elements in the chain which are above the current value of x).

\[ (\text{while } x \neq 3 \text{ do } x := x \oplus 1) = (((x \neq 3) ; x := x \oplus 1)^* ; (x=3)?) \]

Under this perspective, the traditional aspectual verbal classes can be de-
cscribed as distinctive abstract courses of transitions, with the telic/atelic-
distinction corresponding to the terminating/non-terminating property of a
program.

Moreover, if one takes the finite computational sequences or traces of
such programs, one gets a regular set of strings. From a declarative point
of view, as taken by Fernando, verbal meanings then become regular lan-
guages. The symbols of such languages can be thought of as enumerating
propositions which depict a possible state in the model. The strings, accord-
ingly, are recording observations over discrete time, like motion pictures or
comic stripes being accepted by cameras, which can be formulated as finite
automata or finite Kripke models with partial valuations.

On that basis, a regular operation superposition (\&c) over languages (L
and L′) can be introduced as a compositional device for stepwise construction
of complex situational descriptions.

\[ L \& L' = \bigcup_{k \geq 1} \{ (\sigma_1 \cup \sigma_k^1) \ldots (\sigma_k \cup \sigma_k^k) \mid \sigma_1 \ldots \sigma_k \in L, \sigma_k^1 \ldots \sigma_k^k \in L' \} \]

Superposition induces a pre-order subsume (⪰) which, intuitively, compares
two languages according to their informational content.

\[ L \geq L' \text{ iff } L \subseteq L \& L' \]
A simple example (by Fernando), deriving the representation $\Lambda(L)$ for the language “rain from dawn to dusk”, should make the general idea clear. ($\emptyset$ stands for $\theta$-as-a-symbol, boxes replacing braces.)

\[
\text{rain}^+ \& \text{dawn}^+ \& \text{dusk}^+ = \text{rain dawn rain dusk}
\]

Aspectual classes can now be characterized by using a small number of abstract notions defined with respect to the symbols $(\alpha, \omega)$ that start and finish a given language, respectively (where $\sigma$ finishes $L$ if $L \succeq \neg \sigma^+ \sigma$ and $\sigma$ starts $L$ if $L \succeq \sigma \neg \sigma$).

- $\text{telic} (L) = \neg\omega(L)^+$
- $\text{iter} (L) = \emptyset \omega(L)^+$
- $\text{prog} (L) = \emptyset \neg\alpha(L)^+$
- $\text{reten} (L) = \alpha(L)^+$

A certain situational type $L$ falls under one of the four concepts $\chi$ just in case $L \succeq \chi$. The two conditions $\alpha$ and $\omega$ can either be preserved or immediately switched (reading the string from left to right for $\alpha$ and from right to left for $\omega$), which gives a nice encoding of the idea of a situation being initially or finally bounded or unbounded. Below, I marked this property by using a short binary code, with the first digit corresponding to the beginning, the second to the ending, and 1 and 0 indicating the presence or absence of a boundary, respectively.

- $\text{state: reten, iter (0 0)}$
- $\text{activity: prog, iter (1 0)}$
- $\text{achievement: reten, telic (0 1)}$
- $\text{accomplishment: prog, telic (1 1)}$

The following two translations, which give formalizations within the framework of the accomplishment “walk a mile” and the activity “swim” from the initial examples, may serve as an illustration.

\[
(4a) \Lambda(\text{walk a mile}) = \alpha \neg\omega \neg\alpha \neg\omega + = \\
\neg\exists x \leq m (\text{walk}(m)), \neg\text{walk}(m) \exists x \leq m (\text{walk}(m)), \neg\text{walk}(m) + \\
\exists x \leq m (\text{walk}(m)), \text{walk}(m)
\]

\[
(4b) \Lambda(\text{swim}) = \alpha \neg\omega \neg\alpha \omega + = \\
\neg\exists x \neq \emptyset (\text{swim}(x)) \exists x \neq \emptyset (\text{swim}(x)) +
\]
4 The Proposal: Supervaluations

Within the formalism introduced so far, the commonly assumed constraint on the interpretability of temporal adverbials reads as follows (with V denoting a verbal phrase, and I a temporal interval):

\[ \Lambda(V \text{ in } I) = \{ \Lambda(V) \& \Lambda(I) \text{ if } \Lambda(V) \text{ is telic}; \emptyset \text{ otherwise} \} \]
\[ \Lambda(V \text{ for } I) = \{ \Lambda(V) \& \Lambda(I) \text{ if } \Lambda(V) \text{ is iter}; \emptyset \text{ otherwise} \}. \]

The aim of this section is to improve on that by giving room, within the semantic framework itself, for those kinds of derived readings that actually can appear, as shown by the initial examples.

The heart of my proposal is Supervaluation Theory, originally introduced by van Fraassen [21] as a formal tool for handling presupposition failure. The general aim of van Fraassen’s theory was to account for the “third possibility” beside the classical valuations Truth and False in a way that preserves as much as possible from the classical framework. So, notably, while the supervaluationist denies the metalogical principle of bivalence, he still accepts the logical law of excluded middle together with all other classical tautologies. This decisive difference to the several three-valued logical systems known is due to the idea of using truth value gaps rather than a proper third value. Ordinary partial valuations are extended to supervaluations by considering the possible completions of a given model, that is, the set of classical valuations which eliminate all truth value gaps. Take metavariable M to stand for partial models, M’ to range over all possible completions of M, and M* to be the supermodel of M, comprising M together with all its M’.

A supervaluation based on M is a function that assigns Truth with respect to M* to a proposition p just in case p is classically true in all M’, False just in case it is false in all M’, and # (gap) otherwise.

This said, let me now turn back to the problem of giving an appropriately flexible formal characterization of the semantic contribution of temporal adverbials. The algorithm I want to introduce proceeds in three steps, illustrated here by means of example (2) (“walk a mile *for an hour / in an hour”). I assume the semantic representation of the temporal prepositions “in” and “for” under (5a) and (5b) as the starting point. These representations take into account the known sortal preferences relative to the situation modified by encoding the respective properties telic and iter according to the definitions given in the previous section. Prepositional meaning here is semantically characterized in an abstract and context-sensitive but non-ambiguous way.

(5a) \( \Lambda(\text{in}) = \neg \Box \)
(5b) \( \Lambda(\text{for}) = \Box \neg \)

105
With the prepositions having combined with the nominal phrase “an hour” via superposition under (6a) and (6b), the temporal adverbial is now ready to modify the event “walk a mile”, whose abstract characterization is once more given in (7).

\[
(6a) \Lambda(\text{in an hour}) = \neg\omega \uparrow \& \text{time(x)} \uparrow \text{time(y)} \text{ hour(x,y)} = \\
\neg\omega \text{ time(x)} \neg\omega \uparrow \text{ time(y) hour(x,y)}
\]

\[
(6b) \Lambda(\text{for an hour}) = \omega \uparrow \& \text{time(x)} \uparrow \text{time(y)} \text{ hour(x,y)} = \\
\text{time(x)} \omega \uparrow \omega \text{ time(y) hour(x,y)}
\]

\[
(7) \Lambda(\text{walk a mile}) = \alpha \neg\omega \neg\alpha \neg\omega + \omega \neg\alpha \text{ time(y) hour(x,y)}
\]

In the first, obligatory, phase, the representation of the temporal adverbial triggers a copying mechanism, taking the \(\omega\) symbol of the situational string it is going to be combined with, and contributing this symbol or its negation, respectively, to the semantic composition at the positions previously marked by the preposition. If the situational type of the verb phrase is fitting with the structural condition set by the adverbial, this operation does not have any visible effect. The semantic material just combines normally via superposition, as can be seen under (8a), and the algorithm finishes here.

\[
(8a) \Lambda(\text{walk a mile in an hour}) = \\
\neg\omega \alpha \text{ time(x)} \neg\omega \neg\alpha \uparrow \omega \neg\alpha \text{ time(y) hour(x,y)}
\]

However, in case of a sortal clash between the modifying adverbial and the event modified, combining the respective representations in the way described above inevitably leads to a contradiction at some predetermined position inside the complex situational type, so happening in (8b).

\[
(8b) \Lambda(\text{w.a m.*for an hour}) = \neg\omega \alpha \text{ time(x)} \neg\omega \neg\alpha \uparrow \omega \neg\alpha \text{ time(y) hour(x,y)}
\]

These being the cases where interesting aspectual shifts turn up, the real action starts here. In its second phase, the algorithm applies a repairing mechanism by assigning the supervalue \(\#\) to the proposition that previously had received contradictory valuations. The rationale behind this may be thought of as not passing judgement in the face of equally probable but opposing evidences. The result of this operation is shown in (9).

\[
(9) \Lambda(\text{w.a m.*# for an hour}) = \neg\omega \alpha \text{ time(x)} \#\neg\omega \neg\alpha \uparrow \omega \neg\alpha \text{ time(y) hour(x,y)}
\]

The meaning of a temporal adverbial can, accordingly, be thought of as some kind of presupposition of which the semantic contribution is empty in case the verbal concept it combines with is showing the right internal structure, but which introduces a truth value gap at a particular position inside the situational string whenever its structural constraint is not fulfilled.
As stated above, lacking a truth value in the sense of supervaluationism consists in the capacity in principle to make precise in more than one way. That means, for a proposition p having been marked # in a supermodel M*, there are underlying models in M′ such that p is true in one of them, but false in the other. This determination of previously underspecified information in all possible directions by grounding the freshly introduced supervalue # in the underlying classical models, is exactly what the algorithm is supposed to do in its third and last step. So, instead of loosing any information previously received, our ideal language user tries to get the best out of it by developing different hypothetical interpretations separately. Intuitively, this can be taken as a strategy to save monotonicity of the interpretation process by allowing reciprocal adaptions between preposition and situation. This step is spelled out in (10a) and (10b).

\begin{align*}
(10a) & \Lambda^1(w. a m. for an hour) = \neg \omega \alpha \text{ time}(x) \neg \omega \neg \alpha \neg \omega \neg \alpha \neg \omega \neg \alpha \neg \omega \neg \alpha \omega \neg \alpha \text{ time}(y) \text{ hour}(x,y) \\
(10b) & \Lambda^2(w. a m. for an hour) = \neg \omega \alpha \text{ time}(x) \neg \omega \neg \alpha \omega \neg \alpha \text{ time}(y) \text{ hour}(x,y)
\end{align*}

This reintroduction of truth values after a forced gap gives rise to a specific set of new languages. What situational concepts do these rebuilt structures correspond to? In (10a) the preparatory phase of the event is now stretched, leading to a situation where it takes an hour to cross the distance in question. According to the reinterpretation under (10b), on the other hand, the event culminates immediately after start, so that the adverbial can now be interpreted as indicating the duration of the result state\(^4\).

So far so good, but what about the iterative interpretation we found as the third possibility when looking at the example at the very beginning? To get this interpretation derived, let us take one out of the several different-sized strings encoded by the Kleene iteration in the regular expression, which, after step number two of the algorithm, may look like this:

\begin{align*}
(10c)' & \neg \omega \alpha \text{ time}(x) \# \neg \omega \neg \alpha \# \neg \omega \neg \alpha \# \neg \omega \neg \alpha \# \neg \omega \neg \alpha \# \neg \omega \neg \alpha \neg \omega \neg \alpha \text{ time}(y) \text{ hour}(x,y)
\end{align*}

Now, obviously, different hypothetical classical valuations can be chosen for different states, leading, for instance, to the situational pattern under (10c).

\begin{align*}
(10c) & \Lambda^3(walk a mile for an hour) = \\
& \neg \omega \alpha \text{ time}(x) \neg \omega \neg \alpha \neg \omega \neg \alpha \neg \omega \neg \alpha \neg \omega \neg \alpha \neg \omega \neg \alpha \neg \omega \neg \alpha \neg \omega \neg \alpha \neg \omega \neg \alpha \neg \omega \neg \alpha \neg \omega \neg \alpha \text{ time}(y) \text{ hour}(x,y)
\end{align*}

That means, phases of "being-on-the-way" freely alternate with phases of "having-arrived", thereby forming a complex situation which, as a whole, is the attaching point for the temporal information carried by the adverbial.

\(^4\)Even if the result state interpretation is not actually prominent in the particular example considered above, it is certainly present for other accomplishments, as shown by example (3), and therefore needs to be accounted for by the general rule.
The aim of the semantic interpretation process has been achieved at that point. It was to make provision for the set of possible readings found in the cases under consideration in a compositional manner, i.e. only using lexical semantical entries and general rules for combining them. Of course, further methods will have to apply in order to filter out the best actual candidate by the help of more contextual information and general world-knowledge in the pragmatic module of the theoretical language system.

I finish this section by giving the two results within the framework of the analysis of one of the phrases in initial example (1) ("swim in an hour").

\[
\begin{align*}
\neg \omega \alpha & \text{ time}(x) \neg \omega \neg \alpha \\
\theta \omega & \neg \alpha \text{ time}(y) \text{ hour}(x,y)
\end{align*}
\]

5 Conclusion

The aim of this paper was to show how the meaning potential of temporal adverbials can be formally accounted for, and to give room, within the compositional semantic framework itself, for an efficient derivation and systematic explanation of the exact number and kinds of possible readings. According to the proposal made here, aspectual shifts in connection with temporal modification consist in restructuring a situational concept from inside rather than in simple mappings from one atomic event onto another. Aspectual transitions thus happen with respect to representations and are made on a deeper, subatomic conceptual level and by exclusive recurrence to material already present in the enriched dynamic semantic representation. The whole process is triggered indeed by a type conflict, as traditionally assumed, but is now controlled by the introduction and consequent filling in of a truth-value gap. The two relevant factors - aspectual propriety and special ontological relatedness - which appeared formally independent from the point of view of static event semantics and, in the latter case, laid beyond the reach of any known compositional semantic treatment, thus naturally combine inside one and the same simple derivational step. Last but not least, the procedure of systematically introducing underspecification as some kind of claimed ignorance in the presence of conflicting evidences, and of subsequently trying out all possible specifications separately, seems to be a plausible hypothesis about the strategies we really use when adapting sense.

\footnote{The formally possible third reading, parallel to the iterative interpretation given for the previous example, is not very prominent here and normally not discussed in the literature. I, therefore, did not explicitly state it. But there are certainly special contexts in which this condition nevertheless may hold true (take an interval training session, for instance). So, no over-generation here.}
Bibliography


Aspectsual Shift via Supervaluation


Towards An Alternative Proof of Solovay’s Arithmetical Completeness Theorem

THOMAS ICARD
ILLC, Universiteit van Amsterdam
ticardi@emailscience.uva.nl

ABSTRACT. This paper is a first step in an effort to provide an alternative, and, in a sense, more mathematically natural, proof of Robert Solovay’s celebrated Arithmetical Completeness Theorem. We provide a simple axiomatization extending GL for the provability logic of PRA with realizations restricted to a closed fragment, which includes a constant axiomatizing $I\Sigma_1$. The class of Kripke frames for which this logic is sound and complete is none other than the class of frames for GL along with several further properties. Thus we have made one step toward this goal.

1 Introduction

GL is the normal modal logic extending K4 with what is known as Löb’s Axiom, $\Box(\Box A \rightarrow A) \rightarrow \Box A$. This logic is complete with respect to the class of finite, irreflexive, and transitive Kripke frames. One of the most celebrated results in the area of provability logic is Solovay’s Arithmetical Completeness Theorem [Solovay, 1976], which shows that if a sentence $A$ is falsified on some frame in this class, then there is a realization $\ast$, i.e. a function from the set of propositional variables to arithmetical formulas that satisfies certain obvious conditions (e.g. commutativity with the Boolean operations, etc.), such that Peano Arithmetic does not prove $A^\ast$. This result established (along with the more obvious arithmetical soundness) that the class of “always provable” sentences of Peano Arithmetic corresponds in an illuminating way to GL. So, for instance, this gives us an additional tool for finding sentences that, when translated into arithmetical formulas, are not provable in standard number theory. On the other hand, Solovay’s constructive proof, one might argue, is somewhat unnatural. The realization he uses to assure that a given sentence $A$ is not provable in Peano Arithmetic is one that maps each proposition letter to a certain disjunction of sentences. Each one of these sentences asserts, in effect, that the values of a certain function, which itself is obtained using a certain diagonalization
trick, have the sentence’s index as a limit. Ingenious as this proof may be, one might nevertheless hope that there is another, more restrictive realization that could provide us with more mathematically natural sentences.

It is to this larger project that the current paper belongs. The provability logic of another well known number theory, Primitive Recursive Arithmetic (PRA), is also known to be GL. It was proven in [Joosten, 2004] that if we restrict realizations to the fragment with only the provability predicate for PRA, $\top, \bot$, and Boolean combinations of these, this provability logic is complete with respect to the class of irreflexive and transitive linear frames, or, put differently, the class of all GL frames that are non-branching. One might hope that, by strategically adding to this fragment constants that have particular arithmetical interpretations, we could arrive at the class of GL frames with non-triple-branching, then to those with non-quadruple-branching, until, approaching “non-infinite-branching”, we eventually arrive at the class of GL frames tout court. At that point, we would have a new arithmetical completeness proof concerning sufficiently strong arithmetical theories, that makes no use of diagonalization and that promises to be mathematically natural, to the extent that the arithmetical interpretations of these constants are already familiar.

In this paper, we show the provability logic of PRA with realizations restricted to the fragment $F$ below is exactly the logic of GL frames with no triple-branching along with one more requirement we call strict confluence, thus completing the first step towards this goal. We give the following recursive definition of $F$:

$$F := \bot \mid s \mid F \lor F \mid \neg F \mid \Box_{PRA} F$$

where $s$ is the single sentence axiomatizing the theory $I\Sigma_1$, $\Box_{PRA}$ is the provability predicate for PRA (such subscripts will be left out if it is clear we are talking about PRA), and $\bot$ is shorthand for, say, $0 = 1$. Of course $\Diamond_{PRA} F, F \land F$, and so on, are defined in the usual way.

$I\Sigma_1$ is the theory Q (see [Tarski et al., 1953]) along with induction over $\Sigma_1$ formulas. The relationship between $I\Sigma_1$ and PRA is well studied ([Parsons, 1972], [Beklemishev, 1996]). One key fact about this relationship is that $I\Sigma_1$ is $\Pi_2$ conservative over PRA.\footnote{This result has become known as Parson’s Theorem.} This made it possible in [Joosten, 2005] to define a letterless modal logic PGL enriched with a constant for $I\Sigma_1$, that is arithmetically sound and complete with respect to PRA.\footnote{Actually, in addition to Parson’s Theorem, another crucial fact defining PGL is that for all $\Pi_3$ formulas $A$, if $A$ follows in PRA from the negation of the sentence axiomatizing $I\Sigma_1$, then $A$ was already a theorem of PRA.} In fact, PGL is just the modal logic defined by the fragment $F$ and axiomatized by GL plus two extra axioms characterizing the behavior of the constant $s$. A modal semantics is also given (namely it is shown to be
complete with respect to the frame in Definition 1 below), and we shall take advantage of both the arithmetical soundness and modal completeness of PGL.

The provability logic $\text{PL}_F(\text{PRA})$ is thus the set of modal formulas $A$ for which $\text{PRA} \vdash A^*$ for every realization $* \in \text{Sub}(F)$, i.e. for every function from the set of proposition letters to arithmetical sentences in the fragment $F$, that obeys the following restrictions:

1. $\bot^* = \bot$
2. $(\neg A)^* = \neg A^*$
3. $(A \lor B)^* = A^* \lor B^*$
4. $(\Box A)^* = \Box_{\text{PRA}} A^*$

The outline of the paper is as follows: First we provide an axiomatization of a new normal modal logic QGL and show it to be complete with respect to the class of GL frames obeying non-triple-branching and strict confluence.

We shall prove the following theorem:

**Theorem 1.** $\text{QGL} \vdash \varphi$ if and only if $\varphi$ is valid on the class $C$ of frames with the following properties:

1. **Finite**, **Irreflexive** and **Transitive**
2. **Non-triple-branching**: $(Rxy \land Rxz \land Rxw) \rightarrow (Rwy \lor Ryw \lor Rzw \lor Ryz \lor Rzy \lor w = y \lor z = y \lor w = z)$
3. **Strictly Confluent**: $(Rxy \land Rxz \land Ryw) \rightarrow (Rzw \lor Rzw \lor Ryz)$
2.1 Modal Soundness of QGL

First, we shall demonstrate that any theorem of QGL is valid on any frame in $C$ by induction on the complexity of proofs in QGL. Obviously, we have that all K axioms and Löb’s Axiom remain valid on these frames because of P1.

Consider Q1: Suppose for a contradiction we have a state $x$ in $C$ such that $x \models \Box(\Box A \land \neg B \land \neg C) \land \Box(\Box B \land \neg A \land \neg C) \land \Box(\Box C \land \neg A \land \neg B)$. Then there are some $u, v, w$ accessible from $x$ such that $u \models (\Box A \land \neg B \land \neg C)$, $v \models (\Box B \land \neg A \land \neg C)$, and $w \models (\Box C \land \neg A \land \neg B)$. By P2, we must have one of: $Ruv, Ruw, Rwv, Rvw, w = v, w = u, v = u$. Each of these leads to a contradiction.

Next consider Q2: Suppose again for a contradiction that there is a state $x$ in a frame in $C$ such that $x \models \Box(\Box (A \land \Box B) \land \Box C) \land \Box(\Box \neg A \land \neg B \land \neg C)$. Then we have some $y$ such that $Rxy$ and $y \models \Box C$ and $w$ such that $Ryw$ and $w \models (A \land \Box B)$. In addition there is some $z$ such that $Rxz$ and $z \models (\Box \neg A \land \neg B \land \neg C)$. So, by P3, we should have either $Rzw, Rwz, or Ryz$. But each of these again leads to contradiction.

It is obvious that modus ponens and necessitation are valid on any frame in $C$, and we thus conclude the left-to-right direction of Theorem 1.

2.2 Remarks on the Canonical Model for QGL

In order to prove the completeness of QGL with respect to $C$, we shall make use of the canonical model, defined following [Blackburn et al., 2001]:

Definition 2. The canonical model for QGL is the triple $\langle W^{QGL}, R^{QGL}, V^{QGL} \rangle$ such that,

(i) $W^{QGL}$ is the set of all maximal QGL-consistent sets.

(ii) $R^{QGL}wu$ just in case for all formulas $A$, $A \in u$ implies $\Box A \in w$.

(iii) $V^{QGL}(p) = \{w \in W^{QGL} : p \in w\}$

The proof given here is very much in the spirit of, and occasionally directly due to, ([Boolos, 1993]’s version of) [Solovay, 1976]’s modal completeness proof for the logic J (i.e. RGL in [Joosten, 2005]). First we prove some key facts about the canonical model for QGL:

Fact 1. P2 is true of $\langle W^{QGL}, R^{QGL}, V^{QGL} \rangle$.

Suppose not, then for some maximal QGL-consistent sets (henceforth MCS’s), we have $Rxy$, $Rxz$, $Rzw$, $\neg Ryw$, $\neg Rwy$, $\neg Ryz$, $\neg Rzw$, $y \neq z$, $w \neq z$, $w \neq y$. In other words we can find some $A, B, C, D, E, F, G, H$, and $J$ such that $\Box A \in y$, $A \notin w$, $\Box B \in w$, $B \notin y$, $\Box C \in z$, $C \notin w$, $\Box D \in w$, $D \notin z$, $\Box H \in y$, $H \notin z$, $\Box G \in z$, $G \notin y$, $E \in w$, $E \notin z$, $E \notin z$, $E \notin z$, $E \notin z$,
\[ J \in z, J \notin y, \text{ and } F \in y, F \notin w. \] Then let us say, \( K = (D \land B) \lor (E \land \neg F), \)
\( L = A \land H, \) and \( M = (C \land G) \lor \square C \) a subsentence of
\( A, \) so obviously we cannot have \( R_{ww}. \) Finally, we must prove the following
lemma for \( \langle W, R, V \rangle: \)

Fact 2. \( P_3 \) is true of \( \langle W^{QGL}, R^{QGL}, V^{QGL} \rangle. \)

Suppose for a contradiction we have some MCS’s such that \( R_{xy}, R_{xz}, \)
\( R_{yw}, \neg R_{zw}, \neg R_{wz}, \) and \( \neg R_{yz}. \) Then we have some \( A, B, \) and \( C, \) such
that \( \square A \in z, \neg A \in w, \square B \in w, \neg B \in z, \) and \( \square C \in y, \neg C \in z. \) Well, since
\( (\square B \land \neg A) \in w, \) this means \( \Diamond (\square B \land \neg A) \in y, \) but also \( (\square C \land \Diamond (\square B \land \neg A)) \in y, \)
from which it follows \( \Diamond (\square C \land \Diamond (\square B \land \neg A)) \in x. \) On the other hand, the
fact that \( (\square A \land \neg B \land \neg C) \in z \) implies \( \Diamond (\square A \land \neg B \land \neg C) \in x. \) But this
contradicts \( Q_2. \)

2.3 Modal Completeness of \( QGL \)

If \( QGL \not\vdash A, \) then there is some MCS \( u \) such that \( A \notin u. \) If \( \square A \in u, \)
then let us set \( v = u. \) If \( \square A \notin u, \) then \( \neg \square A \in u, \) and by the contra-
positive of L"ob’s Axiom, \( \Diamond (\square A \land \neg A) \in u. \) By the “Existence Lemma”
([Blackburn et al., 2001], p.198) \( (\square A \land \neg A) \in t \) for some \( t \) with \( R^{QGL}_{ut}. \) In
this case let \( v = t. \) Either way we have \( (\square A \land \neg A) \in v. \) Note also that for
any subsentence of \( A, \) of the form \( \square C, \) such that \( \neg \square C \in v, \) we have by the
same reasoning there must be some MCS \( w, \) such that \( R^{QGL}_{uv}, \square C \in w, \)
and \( \neg C \in w. \) Moreover, there are at most two such \( w: \) supposing otherwise,
if we have \( w, \) \( y, \) and \( z, \) all containing \( \square C \) and \( \neg C, \) then by Fact 1, we must
have either one \( R^{QGL}_{uv}-related \) to another, or one equal to the other. Only
the latter of these possibilities will avoid contradiction.

The model we will use to falsify \( A \) is the following:

Definition 3. \( \langle W, R, V \rangle \) is the model for which:

(i) \( W = \{v\} \cup \{x : R^{QGL}_{ux} \text{ and for some subsentence } \square C \text{ of } A, \neg \square C \in v, \neg C \in x, \text{ and } \square C \in x\} \)

(ii) \( R \) is defined as \( R^{QGL} \) restricted to states in \( W \)

(iii) \( V(p) = \{w \in W : p \in w\} \)

Clearly \( \langle W, R, V \rangle \) is finite. It is also clear that \( \langle W, R, V \rangle \) satisfies transi-
tivity and P2 and P3, simply because \( \langle W^{QGL}, R^{QGL}, V^{QGL} \rangle \) does (Facts 1
and 2). And it is immediate that our model also satisfies irreflexivity: for
any \( w \in W, \) we have \( \square C \in w, \) and \( \neg C \in w \) for some \( \square C \) a subsentence of
\( A, \) so obviously we cannot have \( R_{ww}. \) Finally, we must prove the following
lemma for \( \langle W, R, V \rangle: \)
Lemma 1. If $w \in W$ and $B$ is a subsentence of $A$, then $B \in w$ if and only if $w \models B$.

Proof: The basic case, as well as the Boolean cases, are completely straightforward. So we prove only the modal case. Suppose $w \not\models \Box C$. Then for some $x$ such that $Rwx$, we have $x \not\models C$, which by the inductive hypothesis means that $\neg C \in x$, and so because $R^{QGL}wx$, $\Box C \not\in w$.

On the other hand suppose $\Box C \not\in w$. Then $\neg \Box C \in w$, and so $\neg \Box C \in v$ (since if $\Box C \in v$, then because $R^{QGL}vw$, $\Box C \in w$). So for some $x \in W^{QGL}$, $R^{QGL}wx$, $\Box C \in x$, and $\neg C \in x$, so we see that $x \in W$ as well. Obviously $x \neq w$. Since $\neg \Box C \in v$, we know there is some $y$, not necessarily in $W$, such that $R^{QGL}wy$ and $\neg C \in y$. Then by Fact 2 we must have either $R^{QGL}xy$, $R^{QGL}yx$, or $R^{QGL}wx$. It certainly cannot be that $R^{QGL}xy$ because this would mean, since $\Box C \in x$, $C \in y$. If we have $R^{QGL}wx$, then we are done since then $w \not\models \Box C$. And if $R^{QGL}yx$, then by transitivity, we arrive at $R^{QGL}wx$, and thus $Rwx$, and once again we have that $w \not\models \Box C$.

So to conclude the proof of Theorem 1, we see if $A \not\in u$, then $A \not\in v$, and by Lemma 1, $(W, R, V), v \not\models A$. ⊣

3 The Class $C$ and the Frame $\mathcal{M}$

To prove that QGL is the logic of the frame $\mathcal{M}$ we use the following proposition:

Proposition 1. If $F \in C$, then for any point $w \in F$, there exists some point $(m, i) \in \mathcal{M}$, such that there is a bounded morphism from the subframe generated by $(m, i)$ to that generated by $w$.

Proof: We proceed by induction on the number of points in a frame. In the basic case, where we have a frame with one point, say $w$, simply consider the subframe generated by $(0, 0)$ and we have $f = \{(0, 0), w\}$, which is clearly a bounded morphism.

Now suppose we have a frame in $C$ with $n + 1$ points. And take the frame generated by some point $w$. First, we point out that there are only three cases to check: where $w$ has no $R^2$-successor, where $w$ has one “immediate” $R^2$-successor (i.e. where there is but one point $v$ such that $R^2 wv$, and for all $u$ such that $u \neq v$ and $R^2 vu$, we have $R^2 vu$), and where $w$ has two “immediate” successors (this case being analogous to the second). For, if we had three immediate $R^2$-successors, that is, three points none of which were equal or $R^2$-related to any other, this would contradict P2. So we proceed

\footnote{For definitions, and explanation, of the notions of generated subframe and bounded morphism, see, e.g. [Blackburn et al., 2001].}
with the three cases:

**Case I:** Immediate and perfectly analogous to the basic case.

**Case II:** If \( w \) has only one immediate \( R^F \)-successor, say \( v \), then consider the subframe generated by \( v \). Since this frame is clearly still in \( \mathcal{C} \), and it contains \( n \) or fewer elements, we get by the inductive hypothesis that there is a bounded morphism \( f \) from the subframe generated by some \( \langle m, i \rangle \in \mathcal{M} \) to the subframe generated by \( v \). But, if we add two points \( \langle m + 1, 0 \rangle \) and \( \langle m, 1 - i \rangle \) to the subframe generated by \( \langle m, i \rangle \) and add to our bounded morphism the ordered pairs \( \langle \langle m, 1 - i \rangle, v \rangle \) and \( \langle \langle m + 1, 0 \rangle, w \rangle \), it is clear we still have a bounded morphism and we are done.

**Case III:** Let us say \( w \) has immediate successors \( v \) and \( y \), that is, \( R^F wv, R^F wy, v \neq y \), and for all \( t \), such that \( t \neq v \) and \( t \neq y \), if \( R^F wt \) then \( R^F vt \) and \( R^F yt \).

First, we point out that, since our frame is irreflexive and finite (specifically it contains \( n + 1 \) points), we know there is at least one “maximal” point in the subframe generated by \( w \), where “maximal” is defined as follows (I am now using \( R \) in place of \( R^F \) in the generated subframe):

**Definition 4.** A point \( x \) is maximal in a frame generated by another point \( w \) if and only if \( Rwx \) but \( \neg \exists t \) such that \( Rxt \).

It is also the case that there are not more than two such maximal points, for this would contradict P2. And in the event there are two maximal points, we have the following:

**Fact 3.** Say \( x \) and \( y \) are maximal in the subframe generated by \( w \). Then for all \( z \) such that \( Rzw \), if \( z \neq x \) and \( z \neq y \), then \( Rzx \) and \( Rzy \).

For, supposing otherwise, if \( \neg Rzx \) and \( \neg Rzy \), then we have an immediate contradiction to P2. So, without loss of generality, suppose there is a \( v \) such that \( Rvx \) but \( \neg Rvy \), then we have \( Rwv, Rwy, \) and \( Rxv \). By P3, we should either have \( Ryx, Rxy, \) or \( Rvy \). But we have supposed that none of these is the case. Hence \( x \) and \( y \) are truly “equally maximal”.

Now, if we have one maximal point, say \( x \), consider the frame isomorphic to the subframe generated by \( w \), except without the point \( x \). Since all points in the generated subframe were \( R \)-related to \( x \), and our frame is irreflexive, it is clear that if \( F \in \mathcal{C} \), then so is this generated subframe minus \( x \). So by inductive hypothesis, since the frame has \( n \) or fewer points, we have a bounded morphism \( f \) from a generated subframe of \( \mathcal{M} \) to this frame. Now we alter this function \( f \) as follows (adjusting the frame as needed): \( f' = \{ \langle \langle m + 1, i \rangle, z \rangle : \langle \langle m, i \rangle, z \rangle \in f \} \cup \{ \langle \langle 0, 0 \rangle, x \rangle, \langle \langle 0, 1 \rangle, x \rangle \} \)

And this is clearly the bounded morphism we need.

Finally, if there are two “equally maximal” points \( x \) and \( y \), then we consider again the subframe generated by \( w \) minus these points \( x \) and \( y \).
Towards An Alternative Proof of Solovay’s Arithmetical Completeness Theorem

And for the same reason as in the last case (that is, we have by Fact 3 all points in the generated subframe are $R$-related to $x$ and to $y$, but neither is $R$-related to any other point), this frame is also in $C$. And it also has fewer than $n$ elements. So if we have the bounded morphism $f$ from some generated subframe of $M$ to this subframe minus $x$ and $y$, then we let $f' = \{\langle m+1, i \rangle, z \rangle : \langle m, i \rangle, z \rangle \in f \} \cup \{\langle 0, 0 \rangle, x \rangle, \langle 0, 1 \rangle, y \rangle \}$. And this is once again the bounded morphism we need.

We also have the following fact:

**Fact 4.** If there is a bounded morphism from $F$ to $F'$, then there being a $V'$ and $w'$ such that $\langle F', V' \rangle, w' \not\models A$ implies there are $V$ and $w$ such that $\langle F, V \rangle, w \not\models A$.

Finally we can prove the following corollary:

**Corollary 1.** $QGL = \{A : M \models A\}$

**Proof:** Suppose $B \not\in \{A : M \models A\}$. Then there is some $V$ and $\langle m, i \rangle$ such that $\langle M, V \rangle, \langle m, i \rangle \not\models B$. But consider the subframe generated by $\langle m, i \rangle$: This frame is obviously in $C$. And if we let $V'$ be $V$ restricted to points in the subframe, we have a frame in $C$ falsifying $B$. So $B \not\in QGL$.

On the other hand, if $B \not\in QGL$, then we have a frame $F \in C$, valuation $V$, and state $w$, such that $\langle F, V \rangle, w \not\models B$. Then by Fact 4 we know there is some $\langle m, i \rangle \in M$, and valuation $V'$, such that $\langle M, V' \rangle, \langle m, i \rangle \not\models B$. Extending this valuation by, say, making all variables false everywhere else gives us that $B \not\in \{A : M \models A\}$.

$\square$

**4 PL_F(PRA) = QGL**

We are now able to state and prove our central theorem:

**Theorem 2.** $PL_F(PRA) = QGL$

For the right-to-left inclusion, we use the facts mentioned in Section 1 to the effect that the logic PGL is arithmetically sound and modally complete. If $\exists^* \in Sub(F)$ such that $\text{PRA} \not\models A^*$, then if we see $A^*$ not as an arithmetical formula, but as a modal formula, it is proven in [Joosten, 2005] that this implies there is some point $\langle m, i \rangle$ such that $\langle M, V \rangle, \langle m, i \rangle \not\models A^*$ for a particular valuation $V$ in the language of PGL. However, this is no matter as we can convert this valuation into a valuation in the standard modal language by setting for all proposition letters, $V'(p) = \{\langle m, i \rangle : \langle M, V \rangle, \langle m, i \rangle \models p \}$. A simple induction on the complexity of formulas shows that $\langle M, V', \langle m, i \rangle \models p \}$.

$\square$

4 Again, see [Blackburn et al., 2001].

118
By Corollary 1, we conclude that QGL ⊬ A.

Now suppose QGL ⊬ A. Then by Corollary 1, there is some V and (m, i), such that \( M', V, \langle m, i \rangle \not\models A \). Consider the finite frame \( M' \) generated by \( \langle m, i \rangle \) and the valuation \( V' \) restricted to this frame. Clearly \( M', V', \langle m, i \rangle \not\models A \). In order to find a realization falsifying \( A \) in PRA, we first assign the formulas \( \varphi_{(m, i)} \) to each point \( \langle m, i \rangle \) of \( M' \):

\[
\varphi_{(m, i)} = \square^{m+1} \bot \land \diamondsuit^m \top \land s \quad \text{if } i = 1
\]

\[
\varphi_{(m, i)} = \square^{m+1} \bot \land \diamondsuit^m \top \land \neg s \quad \text{if } i = 0
\]

Then, because PRA is complete with respect to the frame \( M \) (as well as \( M' \)) for formulas in the closed fragment \( F \), the following three facts are obviously true (c.f. [Joosten, 2005]):

1. PRA \( \vdash \varphi_{(m, i)} \rightarrow \neg \varphi_{(n, j)} \) if \( \langle m, i \rangle \neq \langle n, j \rangle \)
2. PRA \( \vdash \varphi_{(m, i)} \rightarrow \Box (\bigvee_{n < m} \varphi_{(n,j)}) \)
3. PRA \( \vdash \varphi_{(m, i)} \rightarrow \wedge_{n < m} \Diamond \varphi_{(n,j)} \)

With these facts we can now prove the following lemma. We define \( * \) as such: \( p^* = \bigvee_{\langle M', V' \rangle, \langle m, i \rangle \models p} \varphi_{(m, i)} \). As \( M' \) is finite, clearly \( * \in \text{Sub}(F) \).

**Lemma 2.** For all \( \langle m, i \rangle \) such that \( m \in \omega \) and \( i \in \{0, 1\} \), and all sentences \( C \), if \( \langle M', V' \rangle, \langle m, i \rangle \models C \) then PRA \( \vdash \varphi_{(m, i)} \rightarrow C^* \), and if \( \langle M', V' \rangle, \langle m, i \rangle \not\models C \) then PRA \( \vdash \varphi_{(m, i)} \rightarrow \neg C^* \).

**Proof:** For the basic case, suppose \( \langle M', V' \rangle, \langle m, i \rangle \models p \). Then as \( p^* \) has \( \varphi_{(m, i)} \) as one of its disjuncts, certainly PRA \( \vdash \varphi_{(m, i)} \rightarrow p^* \). And if \( \langle M', V' \rangle, \langle m, i \rangle \not\models p \), then for each disjunct of \( p^* \), \( \varphi_{(m, i)} \) is distinct from it, so by I above, PRA \( \vdash \varphi_{(m, i)} \rightarrow \neg \varphi_{(n,j)} \) for all such \( \langle n,j \rangle \neq \langle m, i \rangle \). So PRA \( \vdash \varphi_{(m, i)} \rightarrow \neg p^* \).

The Boolean cases are all straightforward, so we consider only the modal case.

Suppose \( \langle M', V' \rangle, \langle m, i \rangle \models \Box B \) for some \( B \). Then for all \( \langle n,j \rangle \), such that \( n < m \), \( \langle M', V' \rangle, \langle n,j \rangle \models B \). So by the inductive hypothesis, PRA \( \vdash \varphi_{(n,j)} \rightarrow B^* \) for each such \( \langle n,j \rangle \). That is, PRA \( \vdash \bigvee_{n < m} \varphi_{(n,j)} \rightarrow B^* \). So PRA \( \vdash \Box (\bigvee_{n < m} \varphi_{(n,j)}) \rightarrow \Box B^* \). By II, PRA \( \vdash \varphi_{(m, i)} \rightarrow \Box (\bigvee_{n < m} \varphi_{(n,j)}) \), so PRA \( \vdash \varphi_{(m, i)} \rightarrow \Box B^* \).

And if \( \langle M', V' \rangle, \langle m, i \rangle \not\models \Box B \), for some \( B \), then for some \( n < m \) and \( j \in \{0, 1\} \), we have that \( \langle M', V' \rangle, \langle n,j \rangle \not\models B \), so by inductive hypothesis PRA \( \vdash \varphi_{(n,j)} \rightarrow \neg B^* \). So PRA \( \vdash B^* \rightarrow \neg \varphi_{(n,j)} \), from which it follows PRA \( \vdash \Box B^* \rightarrow \Box \neg \varphi_{(n,j)} \), and thus PRA \( \vdash \Diamond \varphi_{(n,j)} \rightarrow \neg \Box B^* \). But we know by III that PRA \( \vdash \varphi_{(m, i)} \rightarrow \Diamond \varphi_{(n,j)} \), so PRA \( \vdash \varphi_{(m, i)} \rightarrow \neg \Box B^* \). ∴
Towards An Alternative Proof of Solovay’s Arithmetical Completeness Theorem

To complete the proof of Theorem 2, from QGL ⊬ A we arrived at \((M', V'), \langle n, j \rangle \neq A\). And now by Lemma 2, we see that, for our particular realization *, we have PRA ⊨ \(\varphi_{(n,j)} \rightarrow \neg A^*\). So also PRA ⊨ \(\diamond \varphi_{(n,j)} \rightarrow \neg \Box A^*\). Furthermore we see that any \(\diamond \varphi_{(n,j)}\) is true, i.e. \(\varphi_{(n,j)}\) is consistent with PRA: Certainly \(\diamond \Box^{n+1} \bot\) is true for any \(n\), because otherwise \(\Box \diamond^{n+1} \top\) is true, but this is equivalent to the false statement that PRA proves its own consistency. Also, \(\diamond^{n+1} \top\) is true for any \(m\), for otherwise we would have \(\Box^{n+1} \bot\), which is obviously false, assuming PRA is true and consistent. Finally, both \(\diamond s\) and \(\diamond \neg s\) are true, because PRA neither proves \(s\) nor \(\neg s\). Again assuming PRA is a true theory, we know \(\diamond \varphi_{(n,j)} \rightarrow \neg \Box A^*\) is true, and hence \(\neg \Box A^*\) is true. That is to say, PRA \(\not\vdash A^*\). \(\dashv\)

5 Conclusion

We have thus determined the provability logic of PRA with realizations restricted to a closed fragment including a constant axiomatizing \(I\Sigma_1\). We have also shown that this is precisely the logic of non-triple-branching GL frames (that are also strictly confluent). The question now is, of course, what to add to this fragment next, if we want to attain the logic of non-quadruple-branching GL frames, etc., and eventually the logic GL? We leave this question open for further research, but it is worth pointing out certain properties we might expect these constants to exhibit, assuming the constants are interpreted as known arithmetical theories.

First of all, any such theory \(T\) must be finitely axiomatizable to be of any use at all. Second, one would assume that any new \(T\) would satisfy some analogue of Parson’s Theorem (as well as the “\(\Pi_3\) conservativity” of the negation of the sentence axiomatizing \(T\) in PRA). After all, these two facts allowed us to capitalize on the link between arithmetic and modality, afforded by PGL. Third, we clearly cannot have that either PRA proves the consistency of \(T\), nor obviously that PRA proves the sentence axiomatizing \(T\). Fourth, we must make sure that it is not the case that both \(T\) proves \(I\Sigma_1\) and \(I\Sigma_1\) proves \(T\), or else our adaptation of the frame \(M\) would collapse. For, we could never assign to any branch \(\Box^{n+1} \bot \land \diamond \top \land \neg t \land \neg s\) nor \(\Box^{n+1} \bot \land \diamond \top \land \neg t \land s\). It could, however, happen that, say, \(T \vdash \Sigma_1\) but \(I\Sigma_1 \not\vdash T\), since then we get one further branch: \(\Box^{n+1} \bot \land \diamond \top \land \neg t \land s\). At any rate, these are only meant to be speculative remarks. We postpone these questions to future work.\(^5\)

\(^5\)I would like to acknowledge here my indebtedness to Joost Joosten. He has been extremely generous with his time, ideas, and patience. I have enjoyed working with him and look forward to future projects. I would also like to thank the anonymous reviewer whose comments were quite helpful. Finally, thanks to the organizers of the ESSLLI Student Session for providing the opportunity to present this paper.
Bibliography


Towards An Alternative Proof of Solovay’s Arithmetical Completeness Theorem
Factual Content in Algorithmic Natural Language Semantics

ELENI KALYVIANAKI
Graduate Program in Logic, Algorithms and Computation (MΠΛA)
Department of Mathematics, University of Athens, Greece
ekalyv@math.uoa.gr

Abstract. In the theory of referential intensions, the meaning of a sentence of natural language is modeled by the abstract algorithm that computes its reference. In this work, we introduce factual content, an additional structural notion of meaning which depends on context and models the information that a sentence communicates about the world.

1 Introduction

In (Moschovakis 2006), a theory of structured meaning for natural language is proposed where the fregean sense of a sentence is modeled by an abstract algorithm (referential intension) that computes its reference (Truth or Falsity). The theory is developed into a formal language ($L_\lambda$) which is an extension of Montague’s intensional logic and into which natural language is rendered. Central role in this theory plays a (referential) synonymy relation on all terms of $L_\lambda$ which depends on both their structural and denotational characteristics and distinguishes semantically between true mathematical sentences which (obviously) differ in meaning.

This calculus of synonymy is based primarily on the global meaning of a sentence – that is, its meaning with respect to any possible context of reference (or state or index) but its local meaning at a particular state and thus, local synonymy are also defined. For example, in the theory of referential intensions, the sentences (Jlh): “John loves himself” and (Hlh): “He loves himself” are neither referentially synonymous nor locally synonymous even at a state where “John” and “He” denote the same person. Local synonymy still takes into account the structure of the terms and produces thus interesting results.

In the present work, we introduce a third notion of meaning, the factual content, which depends on context and seeks to capture more or less “the information that a sentence conveys about the world”. Based on that, the
two sentences (Jlh) and (Hlh) at the state described above are shown to be \textit{factually synonymous} accounting for the fact that they communicate the same fact about the world. These three notions of synonymy and their interrelations form thus a framework into which we can describe in a complete way the semantics of natural language.

2 Meaning as referential intension

This section is a brief introduction to the theory of referential intensions presented in detail in (Moschovakis 2006) which models the meaning of a term by the “algorithm” that computes its denotation.

$L_{\lambda}^{\tau}(K)$, the formal language into which the theory is developed, is a typed $\lambda$-calculus enriched with a recursive construct. Types are defined recursively by

$$\sigma ::= e \mid t \mid s \mid (\sigma_1 \rightarrow \sigma_2)$$

where the basic types are: $e$ (entities), $t$ (truth values) and $s$ (states).

The set $K$ is a finite set of typed constants $c : \sigma$ which introduce in $L_{\lambda}^{\tau}(K)$ the part of the lexicon of the natural language we are interested in. States are always explicit in the typing of the constants and thus, it is more convenient to think of them as typed in the subset of types of the form

$$\tilde{\sigma} ::= (s \rightarrow e) \mid (s \rightarrow t) \mid \tilde{\sigma} \rightarrow \tilde{\tau}.$$  \hfill (1.2)

Suppose that for our purpose here the set $K$ includes the constants John : $\tilde{e}$, love : $\tilde{e} \times \tilde{e} \rightarrow \tilde{t}$ and $\Box : \tilde{t} \rightarrow \tilde{t}$.

In this language, apart from the usual typed quantifiable variables (called \textit{pure}), there are also \textit{recursive} variables with a distinct role made clear in the recursive construct. The terms of $L_{\lambda}^{\tau}(K)$ are defined by

$$A ::= c \mid x \mid B(C) \mid \lambda(v)B \mid A_0 \text{ where } \{p_1 := A_1, \ldots, p_n := A_n\}$$ \hfill (1.3)

where $x$ is a pure or recursive variable, $v$ is a pure variable and $p_1, \ldots, p_n$ are recursive variables. A type is assigned to each term by this definition and free and bound occurrences of the variables are determined. In what concerns the recursive term, we just note that

(i) $p_1, \ldots, p_n$ are distinct recursive variables and they occur bound in it,

(ii) for all $i = 1, \ldots, n$, $A_i, p_i : \sigma_i$ and if $A_0 : \sigma$, then $A_0$ \textit{where} $\{p_1 := A_1, \ldots, p_n := A_n\} : \sigma$ as well, and finally,

(iii) the system $\{p_1 := A_1, \ldots, p_n := A_n\}$ is \textit{acyclic} - that is, we can associate a natural number to each recursive variable $p_i$, rank($p_i$), so that if $p_j$ occurs free in $A_i$, then rank($p_i$) > rank($p_j$).
For example, the sentences of the natural language (Jlh) and (□ Jlh): “It is necessary that John loves himself” are rendered in $L_{\lambda r}(K)$ as the following terms of type $\tau$:

$$\lambda(x)(\text{love}(x, x))(\text{John}) \quad \text{and} \quad \Box(\lambda(x)(\text{love}(x, x))(\text{John})).$$

An interpretation structure $\mathcal{U}$ of $L_{\lambda r}(K)$ comprises:

(i) non empty sets for the basic types $T_e, T_t$ and $T_s$ and for objects of type $(\sigma_1 \rightarrow \sigma_2)$, the set of functions $T_{\sigma_1 \rightarrow \sigma_2} = \{ f : T_{\sigma_1} \rightarrow T_{\sigma_2} \}$,

(ii) an object $c \in T_{\sigma}$ for every constant $c : \sigma$ of $K$,

(iii) a denotation function $\text{den}$ that associates to each term $A : \sigma$ and each assignment $g$ to the variables, the object $\text{den}(A)(g) \in T_{\sigma}$ which we define here only in the case of the recursive term.

$$\text{den}(A_0 \text{ where } \{p_1 := A_1, \ldots, p_n := A_n\})(g) = \text{den}(A_0)(g\{p_1 := P_1, \ldots, p_n := P_n\})$$

where for $i = 1, \ldots, n$ and if $p_{j_1}, \ldots, p_{j_m}$ are the recursive variables with ranks lower than rank($p_i$), each object $P_i$ is defined by recursion on rank($p_i$) by

$$P_i = \text{den}(A_i)(g\{p_{j_1} := P_{j_1}, \ldots, p_{j_m} := P_{j_m}\}).$$

Apart from the usual denotational semantics, we associate with each proper\(^1\) term $A : \sigma$ its referential intension $\text{int}(A)$ which models its global meaning in $L_{\lambda r}(K)$. In order to do that, a reduction relation ($A \Rightarrow B$) on terms is defined by ten reduction rules which we do not present here and it is shown that for each term $A$, there is a unique (up to congruence) irreducible recursive term

$$\text{cf}(A) \equiv A_0 \text{ where } \{p_1 := A_1, \ldots, p_n := A_n\}$$

such that $A \Rightarrow \text{cf}(A)$. For example,

$$\text{cf}(\lambda(x)(\text{love}(x, x))(\text{John})) \equiv \lambda(x)(\text{love}(x, x))(p) \text{ where } \{p := \text{John}\}.$$ 

The canonical form $\text{cf}(A)$ can be effectively computed from $A$, it is denotationally equivalent with $A$, and its parts $A_0, \ldots, A_n$ determine the basic computable modules that are needed in order to compute the denotation of the original term $A$. Now, the referential intension of $A$ is the tuple of functions

$$\text{int}(A) = (f_0, f_1, \ldots, f_n)$$

\(^1\)In $L_{\lambda r}(K)$, improper or immediate terms are the variables and some simple, variables-like terms (with no constants) which are not assigned meanings.
defined by the parts of its canonical form,
\[ f_i(d_1, \ldots, d_n, g) = \text{den}(A_i)(g\{p_1 := d_1, \ldots, p_n := d_n\}) \quad (i = 0, \ldots, n) \]
and it models the meaning of \( A \) in this theory — the abstract algorithm that computes its denotation \( \text{den}(A)(g) \). Thus, the canonical form of \( A \) defines formally within the language the meaning of \( A \).

The strictest equivalence relation between recursors — tuples of functions like \( \text{int}(A) \) — is natural isomorphism which in this theory is used to define synonymy between terms. For any two terms \( A \) and \( B \), \( A \) is referentially synonymous with \( B \) (\( A \approx B \)) if and only if
\[ A \Rightarrow \text{cf}(A) \equiv A_0 \text{ where } \{p_1 := A_1, \ldots, p_n := A_n\} \]
\[ B \Rightarrow \text{cf}(B) \equiv B_0 \text{ where } \{p_1 := B_1, \ldots, p_n := B_n\} \]
and for each \( i = 0, \ldots, n \) and all \( g \), \( \text{den}(A_i)(g) = \text{den}(B_i)(g) \).

In (Moschovakis 2006), this synonymy relation is developed into a compositional theory of synonymy with very interesting results. For example, since \( \text{John} \neq \text{he} \) where \( \text{he} : \hat{e} \) is a constant in \( K \), it is proved that
\[ \lambda(x)(\text{love}(x, x))(\text{John}) \neq \lambda(x)(\text{love}(x, x))(\text{he}) \]

To define local meanings, it is convenient to enrich the language with a parameter \( \check{a} \) associated with each state \( a \). The local meaning of a term \( A : \check{t} \) at state \( a \in T_\check{t} \) is then defined by \( \text{cf}(A(\check{a})) \).

Thus, at a state \( b \in T_\check{t} \), if the two constants \( \text{love} \) and \( \text{be_fond_of} : \hat{e} \to \check{t} \) are denotationally equivalent, (that is, for all objects \( f_1, f_2 \in T_\check{t} \), \( \text{love}(f_1, f_2, b) = \text{be_fond_of}(f_1, f_2, b) \)), then the two sentences \((\text{Jlh})\) and \((\text{Jfh}): \text{"John is fond of himself"} \) are locally synonymous at \( b \)
\[ \lambda(x)(\text{love}(x, x))(p)(\check{b}) \text{ where } \{p := \text{John}\} \]
\[ \simeq \lambda(x)(\text{be_fond_of}(x, x))(p)(\check{b}) \text{ where } \{p := \text{John}\} , \]
while even if, at a state \( a \), \( \text{John}(a) = \text{he}(a) \), the two sentences \((\text{Jlh})\) and \((\text{Hlh})\) are not locally synonymous at \( a \)
\[ \lambda(x)(\text{love}(x, x))(p)(\check{a}) \text{ where } \{p := \text{John}\} \]
\[ \neq \lambda(x)(\text{love}(x, x))(p)(\check{a}) \text{ where } \{p := \text{he}\}. \quad (1.4) \]

Thus, much like (global) referential synonymy, local synonymy at a fixed state \( a \) still makes fine distinctions between the terms of \( L_\check{a}(K) \) in a non-trivial way that is not determined by the denotations of their parts at \( a \). Nevertheless, the two sentences in (1.4) describe (with different words) the same fact about the world at a particular context and that is what we seek to capture with the alternative situated meaning that we will define in the following two sections.
3 Factual content in LIL

In (Kalyvianaki and Moschovakis 2006), two aspects of situated meaning are introduced for the Language of Intensional Logic of Montague (LIL). We will present here the basic ideas in order to show how factual content and local meaning interrelate and lead to useful insights for problems about indexicals, belief contexts and translation.

LIL (Montague 1973) is a typed λ-calculus whose characteristic is that there are no variables over the type s. In the interpretation structures of this language, the denotation of any term \( A : \sigma \) at an assignment \( g \) is an object \( \text{den}_{\text{LIL}}(A)(g) : T_s \rightarrow T_\sigma \) and the terms \( \cdot(A) \) and \( \cdot(A) \) for each term \( A \) permit us to “express” formally within the language the denotation of \( A \).

The constants that we used in the examples in Section 2 are now typed as \( \text{John} : e \), \( \text{love} : e \times e \rightarrow t \) (an extensional transitive verb) and \( \Box : (s \rightarrow t) \rightarrow t \) (an intensional operator) and the sentences \((\text{Jlh})\) and \((\Box \text{Jlh})\) are rendered as

\[\lambda(x)(\text{love}(x,x))(\text{John}) \quad \text{and} \quad \Box(\cdot(\lambda(x)(\text{love}(x,x))(\text{John}))).\]

On the other hand, \( \text{Ty}_2 \), presented in (Gallin 1975), admits all types and is denotationally equivalent with LIL. The following tables summarize the way the two languages are related.

<table>
<thead>
<tr>
<th>Types</th>
<th>( \sigma \equiv e \mid t \mid (s \rightarrow \sigma_2) \mid (\sigma_1 \rightarrow \sigma_2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terms</td>
<td>( A \equiv x \mid c \mid A(B) \mid \lambda(x)(B) \mid \cdot(A) \mid \cdot(A) )</td>
</tr>
<tr>
<td>Denotation of ( A : \sigma )</td>
<td>( \text{den}<em>{\text{LIL}}(A)(g) : T_s \rightarrow T</em>\sigma )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Types</th>
<th>( \sigma \equiv e \mid t \mid s \mid (\sigma_1 \rightarrow \sigma_2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terms</td>
<td>( A \equiv x \mid c^G \mid A(B) \mid \lambda(x)(B) )</td>
</tr>
<tr>
<td>Denotation of ( A : \sigma )</td>
<td>( \text{den}(A)(g) : T_\sigma )</td>
</tr>
</tbody>
</table>

The Gallin translation \( A^{G,u} : \sigma \) of a LIL-term \( A : \sigma \) (relative to a state variable \( u \) that stands for “the current state”) is a \( \text{Ty}_2 \)-term such that for all \( g \), if \( g(u) = a \), then \( \text{den}(A^{G,u})(g) = \text{den}_{\text{LIL}}(A)(g)(a) \).

If \( K \) is again the set of constants of LIL, the Gallin translation presupposes that for each constant \( c : \sigma \) in that set, we introduce a new one \( c^G : s \rightarrow \sigma \), forming the set \( K^G \). For example, \( \text{John}^G : s \rightarrow e \) and \( \text{love}^G : s \times e \times e \rightarrow t \) are members of \( K^G \). The LIL-term \( \Box(\cdot(\lambda(x)(\text{love}(x,x))(\text{John}))) \) is translated in \( \text{Ty}_2 \) as

\[\Box(\cdot(\lambda(x)(\text{love}(x,x))(\text{John}))) \equiv \Box^G(u)(\lambda(u)[\lambda(x)(\text{love}(x,x))(\text{John})]^G,u) \equiv \Box^G(u)(\lambda(u)\lambda(x)(\text{love}^G(u)(x,x))(\text{John}^G(u))).\]
Introducing the recursive construct in Ty₂, L^λ_{ar}(K^G) is formed into which we can define for each proper LIL-term \( A : \sigma \) and each state \( a \in \mathbb{T}_s \) the following:

- **Factual content of \( A \) at state \( a \):** \( FC(A, a) = \text{int}(A^{G, \bar{a}}) \)
- **Global referential meaning of \( A \):** \( M(A) = \text{int}(\lambda(A^{G, u})) \)
- **Local meaning of \( A \) at state \( a \):** \( LM(A, a) = \text{int}(\lambda(A^{G, u})(\bar{a})) \)

For example, the factual content and the local meaning of the term \( \lambda(x)(\text{love}(x, x))(\text{John}) \) at any state \( a \) are defined by the corresponding canonical forms

\[
\text{cf}(\lambda(x)(\text{love}^{G}(\bar{a})(x, x))(\text{John}^{G}(\bar{a}))) \\
\equiv \lambda(x)(\text{love}^{G}(\bar{a})(x, x))(p) \text{ where } \{p := \text{John}^{G}(\bar{a})\}
\]

\[
\text{cf}(\lambda(u)(\lambda(x)(\text{love}^{G}(u)(x, x))(\text{John}^{G}(u)))(\bar{a})) \\
\equiv \lambda(u)(\lambda(x)(\text{love}^{G}(u)(x, x))(p(u)))(\bar{a}) \text{ where } \{p := \lambda(u)\text{John}^{G}(u)\}.
\]

We can now define **factual synonymy** between a term \( A \) at state \( a \) and a term \( B \) at state \( b \) as the referential synonymy of their translations,

\[
FC(A, a) = FC(B, b) \iff A^{G, \bar{a}} \approx B^{G, \bar{b}}.
\]

Returning to our familiar example, at a state \( a \in \mathbb{T}_s \) such that \( \text{John}^{G}(a) = \text{he}^{G}(a) \) the two sentences \((\text{Jlh})\) and \((\text{Hlh})\) are factually synonymous,

\[
\lambda(x)(\text{love}^{G}(\bar{a})(x, x))(p) \text{ where } \{p := \text{John}^{G}(\bar{a})\} \\
\approx \lambda(x)(\text{love}^{G}(\bar{a})(x, x))(p) \text{ where } \{p := \text{he}^{G}(\bar{a})\},
\]

while, as in \( L^\lambda_{ar}(K) \), they are not locally synonymous:

\[
\lambda(u)(\lambda(x)(\text{love}^{G}(u)(x, x))(p(u)))(\bar{a}) \text{ where } \{p := \lambda(u)\text{John}^{G}(u)\} \\
\not\approx \lambda(u)(\lambda(x)(\text{love}^{G}(u)(x, x))(p(u)))(\bar{a}) \text{ where } \{p := \lambda(u)\text{he}^{G}(u)\}.
\]

The idea is that \((\text{Jlh})\) and \((\text{Hlh})\) express the same fact about the world in state \( a \), but *they do not mean the same thing in \( a \)—one could rationally believe one without believing the other.*

**Theorem 1** For terms of Montague’s intensional logic, referential synonymy implies local synonymy at every state, and local synonymy at a fixed state \( a \) implies factual synonymy at \( a \).
Neither of the implications in this Theorem can be reversed in general, and the distinctions between these three meanings associated with each LIL-term provoke interesting insights about the semantics of indexicals and belief claims. In particular, in the case of indexicality, it is presented in (Kalyvianaki and Moschovakis 2006) how factual content and global meaning make analogous semantic distinctions as those made by Content and Character defined in the Logic of Demonstratives (Kaplan 1978) and in addition, the role of local meaning in these considerations. In what concerns propositional attitudes, they suggest that the objects of belief are local meanings—not factual contents, as is sometimes assumed, leading to paradox.

4 Factual content in $L^\lambda_{ar}(K)$

In Section 3, the definition of the factual content for LIL-terms exploited the Gallin translation of LIL into $L^\lambda_{ar}$ and the reduction calculus in it. It seems plausible that one can define directly a natural notion of factual content for arbitrary terms of $L^\lambda_{ar}$, moreover, since it is more expressive than LIL, cf. (Moschovakis 2006).

The approach uses a formal characterization of terms (locality) which is based on the way the computation of the denotation of a term depends on a context of reference. It is not possible to present here all the results of this work due to space limitations. Instead we confine our presentation to communicating the basic ideas involved in the case of a subset of the terms of $L^\lambda_{ar}$ (local terms) whose denotation at a state depends only on the denotations of their arguments at that state.

The section ends with a simplified example involving non local terms which depicts the way factual content can be a vehicle of better understanding intensional contexts in an algorithmic semantics of natural language.

4.1 Local terms

Every type as in (1.2) can be “unfolded” in the general form

$$\bar{\sigma} \equiv \bar{\sigma}_1 \times \ldots \times \bar{\sigma}_n \rightarrow \bar{\sigma}_0$$

where $\bar{\sigma}_0 \equiv \bar{e}$ or $\bar{t}$. Thus, it is clearly suggested that any object $f : \bar{\sigma}_1 \times \ldots \times \bar{\sigma}_n \rightarrow \bar{\sigma}_0$ in the interpretation structure of $L^\lambda_{ar}(K)$ has $n$ arguments and its value on appropriately typed arguments $f_1, \ldots, f_n$ and at state $a \in T_s$ is simply an object $f(f_1, \ldots, f_n, a)$ which is either an entity in $T_e$ or a truth value in $T_t$.

Suppose now we consider the transitive verb love : $\bar{e} \times \bar{e} \rightarrow \bar{t}$. Its denotation love is an object of the same type and to say that this object is local on both its first and second argument means intuitively that for any two pairs of arguments $f_1, f_2 : \bar{e}$ and $f'_1, f'_2 : \bar{e}$ and any state $a$, if $f_1(a) = f'_1(a)$
and \( f_2(a) = f'_2(a) \), then \( \text{love}(f_1, f_2, a) = \text{love}(f'_1, f'_2, a) \). The general idea is that an object \( f : \hat{\sigma}_1 \times \ldots \times \hat{\sigma}_n \rightarrow \hat{\sigma}_0 \) is local if its value on arguments \( f_1, \ldots, f_n \) and a state \( a \) depends only on the values of these arguments on that particular state \( a \). Formally, an object is local if and only if it has a local associate:

**Definition 2** If \( f : \hat{\sigma} \) or \( \hat{\sigma} \rightarrow \), then \( f_* = f \) is the local associate of itself; and recursively: if \( f : \hat{\sigma}_1 \times \ldots \times \hat{\sigma}_n \rightarrow \hat{\sigma}_0 \), then \( f_* : s \times \sigma_1 \times \ldots \times \sigma_n \rightarrow \sigma_0 \).

is a local associate of it if and only if for any arguments \( f_1 : \hat{\sigma}_1, \ldots, f_n : \hat{\sigma}_n \) and their local associates \( (f_1)_* : s \rightarrow \sigma_i \) and for any \( a : s \),

\[
f(f_1, \ldots, f_n, a) = f_*(a, (f_1)_*(a), \ldots, (f_n)_*(a)).
\]

Naturally enough, all objects of type \( \hat{e} \) and \( \hat{t} \) are local but not all objects of types of higher level are. For example, \( \text{den}(\square) : \hat{t} \rightarrow \hat{t} \) is not local, simply because, for any \( f : T_s \rightarrow T_t \) and any \( a \in T_t \), the object \( \text{den}(\square)(f, a) \) depends on the entire function \( f \) and not just on its value, \( f(a) \).

Notice that we define a local associate of an object \( f \) only on arguments that are themselves local. Thus, an object \( f \) can have many local associates which, however, can only differ on arguments that are not local associates of local objects.

We can easily prove useful properties about local associates, for example that if \( f_1, f_2 \) are local, then so is \( f_1(f_2) \) and \( (f_1(f_2))_* (a) = f_1_*(a, f_2_*(a)) \) for any \( a \in T_s \), and using them, prove by induction on the formation rules of the terms of \( \text{L}_\text{ar}^\lambda (K) \) that

**Lemma 3** If all \( c : \hat{\sigma} \in K \) are such that \( c \in T_s \) are local, then the function

\[
f_A(f_1, \ldots, f_n) = \text{den}(A)(g\{x_1 := f_1, \ldots, x_n := f_n\})
\]

is also local, where \( A : \hat{\sigma} \) is a term of \( \text{L}_\text{ar}^\lambda (K) \) with free variables in the list \( x_1, \ldots, x_n \) and \( f_1, \ldots, f_n \) are any objects of appropriate types.

Now, for each closed term \( A \) such that \( \text{den}(A) \) is a local object, the question is whether its associate \( (\text{den}(A))_* \) can be formally defined. This is true, and we sketch here some of the ideas of the proof.

First we extend \( \text{L}_\text{ar}^\lambda (K) \) by a new formation rule for terms, **associate application**, such that if \( B : \hat{\sigma} \) and \( C : s \), then \( B[C] : \sigma \), and

\[
\text{den}(B[C])(g) = (\text{den}(B)(g))_* (\text{den}(C)(g)).
\]

Using this construct (and much as we did in translating \( \text{LIL} \) into \( \text{L}_\text{ar}^\lambda \)), we can associate with each term \( A : \hat{\sigma} \) and a state variable \( u \), a new term \( A_n^* u : \sigma \) so that the following hold:

\footnote{In this section, \( \hat{\sigma} \) is any type as in (1.1) while \( \sigma \) is any type of the form \( \langle e \mid t \mid (\sigma_1 \rightarrow \sigma_2) \rangle \), that is, where \( s \) is not a basic type anymore.}
(i) For closed $A, A^{*,u}$ defines the associate of $A$ at $u$, that is $\text{den}(A^{*,u})(g\{u := a\}) = (\text{den}(A)(g))_*(a)$.

(ii) The $*$-transformation respects the reduction calculus, that is, $\text{cf}(A^{*,u})$ is congruent with $(\text{cf}(A))^{*,u}$.

This last fact allows us to define a natural notion of factual content for local $L^\lambda_{ar}$-terms, an “algorithm” which naturally computes $\text{den}(A)(a)$ using only the denotations of the parts of $A$ at $a$:

$\text{FC}(A, a) = \text{int}(\text{cf}(A^{*,\bar{a}}))$.

For example, $(\lambda(x)\text{love}(x, x)(\text{John}))^{*,u} \equiv \lambda(y)\text{love}[u](y, y)(\text{John}[u])$ and thus, the factual content of $(\text{Jlh})$ at any state $a$ is defined by

$\lambda(y)\text{love}[\bar{a}](y, y)(p)$ where $\{p := \text{John}[\bar{a}]\}$.

It provides the natural algorithm which decides whether John loves himself at state $a$ from a knowledge of the relation of self-love at $a$ and who John is at that state. The factual content of $(\text{Hlh})$ at any state $a$ is expressed similarly and if at state $a$, $\text{John}(a) = \text{he}(a)$,

$\lambda(y)\text{love}[\bar{a}](y, y)(p)$ where $\{p := \text{John}[\bar{a}]\}$

$\approx \lambda(y)\text{love}[\bar{a}](y, y)(p)$ where $\{p := \text{he}[\bar{a}]\}$,

that is, the two utterances are factually synonymous. The result is analogous to the one obtained by the factual synonymy defined in Section 3.

### 4.2 A non local example

As mentioned already, local objects are just a subset of the objects in an interpretation structure of $L^\lambda_{ar}(K)$. To treat non local terms, a generalized version of an associate of an object is defined and, as in Section 4.1, expressed formally within the language. Part of the complexity involved in the treatment of general terms under this approach is due to the fact that it does not assume any (local or non local) uniform behavior of the terms but rather seeks to trace it on the way the terms are used and formed. Moreover, it even allows the possibility of an object, and thus of a term, to be local on one of its arguments and non local on others.

For example, consider the denotation of the constant $\text{former} : (\bar{e} \rightarrow \bar{t}) \times \bar{e} \rightarrow \bar{t}$. By definition, for objects $f_1 : \bar{e} \rightarrow \bar{t}, f_2 : \bar{e}$ and $a : s$,

$\text{former}(f_1, f_2, a) = 1 \iff f_1(f_2, b) = 1$ where $b$ is any state that differs from $a$ in that it expresses past time. It follows easily that the object $\text{former}$ is local on the second but non local on its first argument.

Finally, we present in what follows a simplified version of an example of factual synonymy between two sentences that involve the (intensional)
necessity operator \( \Box : \tilde{t} \rightarrow \tilde{t} \). Assuming a de dicto interpretation, the factual content of the sentence \((\Box \text{Jlh})\) at a state \(a\) is

\[ \Box[\tilde{a}](p) \quad \text{where} \quad \{p := \lambda(u)\lambda(x)(\text{love}[u](x,x))(q), q := \lambda(u)\text{John}[u]\} \]

and even at a state \(a\) where \(\text{John}(a) = \text{he}(a)\), it will not be factually synonymous with the utterance of \((\Box \text{Hlh})\) at that state, since, simply,

\[ \text{FC}(\text{Hlh}, a) \equiv \Box[\tilde{a}](p) \quad \text{where} \quad \{p := \lambda(u)\lambda(x)(\text{love}[u](x,x))(q), q := \lambda(u)\text{he}[u]\} \]

and \(\text{den}(\lambda(u)\text{John}[u]) \neq \text{den}(\lambda(u)\text{he}[u])\). Given the rendering of the two sentences where \(\Box\) is treated as a sentential operator, it is shown here that part of the computation of the factual content at a particular state is the den\((\lambda(u)\text{John}[u])\) = den\((\text{John})\) and not just its value at that state. The information that this sentence communicates at state \(a\) is not about the object \(\text{John}(a) : e\) but about \(\text{John} : \tilde{e}\).

## 5 Conclusion

We presented here two different approaches in defining a natural, structural, non-trivial factual content in the theory of referential intensions and we explored the relation of factual synonymy that it determines.

In Section 3, factual content and local meaning were defined for terms of \(\text{LIL}\) while in Section 4, factual content was defined directly in \(\text{L}_{\text{ar}}^\Lambda(K)\), mainly for local terms. The case of arbitrary, non local terms in its full generality is part of the future work in this area which primarily aims at providing intensional contexts with semantics in an algorithmic theory of meaning.

## Acknowledgements

I thank my supervisor, Prof. Yiannis N. Moschovakis, for his close and inspiring attention over this work throughout its course. I also thank the anonymous ESSLLI reviewers for their comments that helped finalize the structure of this paper.

The project is co-funded by the European Union - European Social Fund & National Resources - EPEAEK II.

## Bibliography


Applying a Focus Tree Model of Dialogue Context to Interactive Question Answering

MANUEL KIRSCHNER
KRDB Center, Faculty of Computer Science
Free University of Bozen-Bolzano, Italy
kirschner@inf.unibz.it

ABSTRACT. In Interactive Question Answering (IQA), dialogue context is needed to resolve context-dependent discourse phenomena, which occur relatively frequently in these dialogues. A focus tree is one viable model for representing dialogue context. We present a new IQA system which is based on this kind of model, and give a detailed overview of how this system has been built. The resulting system is used both as a practical IQA system that will help users retrieve the information they need, and as a test-bed for studying dialogue context models with real users.

1 Introduction

Lately, there has been increasing interest in how to best enrich QA applications with dialogue capabilities\(^1\). Interactive Question Answering (IQA) allows users to get concise answers to their information needs via cooperative natural-language dialogue. While classical QA is concerned with questions posed in isolation, its interactive variant keeps track of the QA process and supports the user in finding the exact solution via natural-language dialogue. In order to do so, it needs to model the dialogue context in which utterances are issued. Context has to be considered for appropriately handling clarification subdialogues, to resolve anaphora, ellipses or fragmentary utterances, and finally, to merge all the information provided over a series of turns, so that an answer to the complex question can be determined. We believe that effective use of context modeling in IQA lags behind. One of the main goals behind our research is to study different models of dialogue context (the focus tree introduced in this paper being one of the possible models). For this research, the general plan is to adopt an empirical approach: implementing

\(^1\)E.g., Workshop on Interactive Question Answering (IQA’06), at HLT-NAACL’06

Proceedings of the Twelfth ESSLLI Student Session
Ville Nurmi and Dmitry Sustretov (editors)
This article, Copyright © 2007, Manuel Kirschner
Applying a Focus Tree Model of Dialogue Context to Interactive Question Answering

different models in a practical IQA system, and then validating them with real user data.

Dialogue context is needed to resolve context-dependent discourse phenomena that occur in dialogues. These phenomena typically include pronouns and anaphoric noun phrases, elided phrases (missing semantic arguments), and fragments. In the course of a series of user utterances within an IQA session, each of these phenomena establishes some kind of dependency between the single utterances. We conducted a Wizard-of-Oz study with librarians of the university library and actual library users (cf. Kirschner (2006) for the experimental setup). One of the goals of this study was to analyze discourse phenomena occurring in actual user log files. Of the initially collected dialogues, around one quarter exhibited some kind of discourse phenomenon. Albeit on the lower end, this ratio is still within the spectrum reported in the literature. Conversation log files of information-seeking tasks in particular have been reported to contain the lowest share of context-dependent turns (Dahlbäck and Jönsson (1989), in Bertomeu et al. (2006)). Interpreting our experimental results, we believe that studying models of dialogue context is worthwhile both from a theoretical point of view, as well as with respect to the practical IQA system that will be described later in this paper.

This paper is structured as follows. In section 2, we introduce the notion of dialogue context, and explain the concept of focus trees. We also provide an overview of some relevant literature. Section 3 explains the background and general design principles of an IQA system that we are developing. Finally, in section 4, we present the implementation in detail.

2 Modeling dialogue context: Previous work

In order to correctly interpret every user utterance from a series, an IQA system needs a model of dialogue context that incorporates context-dependent discourse phenomena. More generally, for every new user utterance, the dialogue context model should correctly predict whether the topic of the interaction has stayed the same or switched to something new (possibly related in a specific way to the previous topic). De Boni and Manandhar (2005) and Yang et al. (2006) describe two approaches to recognizing whether the topic has changed between two subsequent user utterances. In both cases, the decision is based on a set of linguistic features extracted from the utterances; the features are then combined in decision algorithms using heuristics or supervised machine learning, respectively.

While these approaches are effective in terms of detecting topic changes, they do not attempt to model the patterns of topic change. As for these patterns, Bertomeu et al. (2006) provide an empirical study of thematic relations holding between user questions and the preceding context, and of
the location of antecedents between user utterances in IQA dialogues. Different architectures and dialogue theories have been proposed for modeling dialogue context, and to explain certain patterns of topic change. What follows is a review of how focus trees have been used in this respect.

2.1 Modeling changes of dialogue topic using a focus tree

A focus tree provides a way of modeling the dialogue context that can account for topic changes occurring in the course of a dialogue. The main idea is to organize all the topics of the IQA system’s task domain hierarchically. The nodes of this tree represent the current conversational topic (i.e., a concept that has already been mentioned in the ongoing dialogue and that is currently in the focus of the dialogue participants). Representing the current topic via a specific node in the tree is based on the following notion: topic shifts to a somehow “related” topic are more likely than jumping to unrelated ones. In a focus tree, relatedness can be modeled via structural relations between nodes of the tree.

Several different ways of designing such trees have been mentioned in the literature (generally without giving a formulation of some rigorous algorithm). The applications for which focus trees have been employed are varied. McCoy and Cheng (1991) use the tree to constrain what should be said next in a natural-language generation system, by representing the cognitive load of different topic shifts in the tree. Jokinen et al. (1998) start with a manually built tree for marking up main topics in task-based dialogue data, which is extended by an n-gram-based model for topic shifts. The application: predicting the next topic in a spoken dialogue system. Finally, Stede and Schlangen (2004) propose to use a focus tree (in the form of a LOOM taxonomy) for dialogue management; given a user’s dialogue act, the system retrieves a reply from the taxonomy based (at least partly) on structural aspects of the tree.

As in the three approaches just mentioned, the focus tree we are using in our IQA implementation (introduced in section 4) is also built entirely by hand. Thus, it relies critically on the exact way the tree was constructed. While this seems to be a more general problem with knowledge-intensive NLP systems, we hope to alleviate it in the future by defining some formal requirements for the construction of focus trees. Another way of avoiding the uncertainty of building focus trees by hand is to try to learn them from data. To point to one data-driven alternative: Niwa et al. (1997) learn certain relations between topics from free text. However, in our case, the lack of large amounts of training data prevents us from using such data-driven approaches.
2.2 Extended system interactivity by exploiting the dialogue context model

Besides the question of how to best model the dialogue context, another interesting issue for research has been the role of dialogue management in IQA. Here, we are concerned with the general dialogue strategy that the system should adhere to in a conversation (see Core et al. (2003) for a comparison of dialogue strategies in the context of tutorial dialogue). More specifically, one should identify the most helpful system responses at any point in the IQA dialogue. For example, Varges et al. (2006) describe a system that can modify the constraints of a user query by engaging in clarification subdialogues. A further goal of extended system interactivity could be to let the system actively guide the user through the information seeking process.

It is an open question whether an IQA system can provide certain types of extended interactivity by exploiting its model of dialogue context. The underlying notion is to use the dialogue context as a source for supporting (meta) knowledge that can be communicated to the user via system initiated turns. Thus, the system would not only answer user questions from within the task domain using the structured knowledge source, but also implicit knowledge extracted directly from the current state of the context model. The idea is that the user might benefit from viewing some version of contextual information that is normally not visible to him. An interesting starting point in this direction is provided by Chai and Jin (2004). As an IQA dialogue evolves and grows longer, they build up rich contextual information in the form of a directed acyclic graph that encodes the discourse roles and discourse relations introduced so far; they conjecture that these graphs could be used also as a basis for collaborative QA.

3 Proposed approach

For our study of dialogue context in IQA, we have been adopting a bottom-up approach: we start by implementing a baseline system that, while still being rather simplistic regarding the underlying theories, works robustly for a large proportion of the use cases. Talking about a practical IQA system, we start with a shallow natural language understanding component (namely regular expression pattern matching), to do the mapping between user queries and system responses. The initial model for dialogue context is a focus tree, which provides a simple (and arguably too limited) way of keeping the dialogue state between two user turns. The type of interaction in our baseline IQA system is limited to a user-initiated stimulus-response loop, i.e., there is no system initiative yet, but the system simply returns one fixed response for every user utterance it receives. As soon as this system will be running, we propose to start collecting real data (from user interactions). Under the bottom-up paradigm, we expect to gain insights
by looking at such data; these insights should guide us to the next most important aspect of our baseline system that should be fixed or gradually upgraded to a more sophisticated model. We have started implementing these principles for designing a practical IQA system as a case study, which we now introduce.

3.1 Case study

Our university library is striving to improve their on-line information services. An IQA system provides permanent and instant access to library-specific information. As the experiences of other libraries have shown, such systems can surpass static information resources like FAQ lists in that they guide users towards a solution when initially they did not know the exact question.

Together with a team of librarians, we have started building a practical IQA system named BoB (the Bolzano library Bot). This project serves as a case study for implementing theories of dialogue context in IQA, and for validating them with real user data. As our research project (and our implementation of BoB) evolves, the library will have at their disposal an increasingly powerful IQA system. One of the long-term goals of the project is to support information seeking dialogues in three languages (English, Italian and German). See Kirschner (2007) for an overview of dedicated software tools that the librarians and domain experts use for the administration and translation of BoB’s knowledge base from German into the other two target languages.

4 Implementation of a practical IQA system

We will now describe some results in terms of the current implementation of BoB, a practical IQA system for our university library. We start by showing how we built the focus tree from hand-coded data that we imported from another system. We then elaborate on the current implementation of BoB, describing in detail how it uses the library domain focus tree to yield a baseline IQA system.

4.1 A focus tree for the library domain

Through a cooperation with the library of the University of Hamburg, we acquired the library domain knowledge base2 of Stella, a “chatterbot” (simple text-based dialogue system) implementation based on proprietary code. We planned to use these data for two purposes: to jump-start the creation of a focus tree for our own university library domain, and to extract and

---

2Note that the term “knowledge base” is used informally here; it refers to a hierarchy of library topics whose structure is not formalized.
re-use as much as possible of the information that was encoded over several years by a team of librarians in Hamburg.

With respect to jump-starting the creation of our own focus tree, we considered the Hamburg data to be interesting because of the following properties. Firstly, the application domain is very similar to ours: both Hamburg’s chatterbot and our proposed IQA system provide a wide array of support to the users of a university library. Also, their knowledge base has been created, extended and fine-tuned by a team of around five librarians over several years. As a consequence, we hope that the quality and quantity of library topics encoded in the knowledge base let us build our first baseline IQA system with a good coverage of library-related questions and answers. Although we doubt that the Hamburg knowledge base will serve us directly for reaching new insights about dialogue context modeling, we do expect that from a data-driven perspective to building dialogue systems, the more data we have, the better.

The Hamburg library knowledge base encodes 230 topics in a focus tree. As stated above, our secondary goal for incorporating the Hamburg data was to extract and take advantage of as much hand-coded information as possible that had been entered by Hamburg’s librarians. Looking inside the 230 topics, the knowledge base consists of an overall of over 2000 pairs of 1. a regular expression pattern to match some user input, and 2. a canned-text system response to be returned to the user.

We do not know the exact principles with which Hamburg’s focus tree was constructed, but after looking at their knowledge base in some detail, we conjecture that their librarians mixed different principles of organizing topics into a hierarchy. By analyzing the data in detail, i.e., on the source code level, it turned out that, besides containing the previously mentioned regular expression patterns plus system responses organized into the focus tree, they contain a host of additional information, some of which will be described in section 4.2. In the next section, we describe our current implementation of BoB, explaining in detail the basic focus tree search algorithm.

### 4.2 The BoB system

We have implemented BoB as a Java-based web application that will eventually be deployed on the library web site. Using the focus tree as a model for dialogue context, the system can in principle process user utterances that contain certain discourse phenomena (i.e., the above mentioned fragments,)
ellipses and anaphora). What follows is a description of the underlying notion behind how BoB uses a focus tree to represent the dialogue context, and to generate a system response to a user query. Like most chatterbots in the tradition of ELIZA (Weizenbaum (1966)), our system is based on a stimulus-response loop for mapping a user utterance to some corresponding answer. All responses are stored as canned-text strings. Responding to some user query is thus a problem of identifying the best response, which is then simply output to the user. The mapping from user input to system response is done on the basis of regular expression patterns; for every system response, we have stored a regular expression pattern that matches certain types of user input.

In BoB, each regular expression pattern for matching user input is stored in combination with a pre-canned system response. Unlike in most chatterbots, these pairs are organized hierarchically as nodes of a focus tree, where each node represents a specific dialogue context. In the course of a dialogue, the current topic switches between the nodes of the tree, depending on what regular expression patterns the current user utterance matches, and at which node the search for a matching regular expression pattern starts. In this simple model of dialogue history, the current focus node represents the dialogue state, i.e., it encodes all the information that is preserved between two succeeding user utterances.

As mentioned above, the knowledge base we acquired from Hamburg contains a host of information that goes beyond the topic hierarchy and the regular expression patterns and canned-text answers encoded in each focus node. Some parts of this additional information seemed too idiosyncratic to re-use for the BoB system. For example, some nodes in the Hamburg focus tree contain hand-tuned weights for changing the precedence in which they are processed by the search algorithm. Since we do not know how these weights were chosen, nor how exactly the original search algorithm uses them, we did not consider them in our system. On the other hand, we do re-use two extra features encoded in Hamburg’s focus tree, namely context-dependent follow-up questions and system-initiated subdialogues.

Re-using context-dependent follow-up questions

One of the distinguishing features of Interactive QA is that it allows users to pose questions that are related in certain ways to the previous dialogue. We call every question in an IQA dialogue “follow-up” if there exists at least some previous user question or system response, since all follow-ups are potentially related to the dialogue context. An analysis of the kinds of (thematic) relations that may hold in these situations is outside the scope of this paper, but we believe it to be important to further the understanding of IQA dialogues in general, and the requirements for practical IQA systems (manuscript in preparation). At this point, we are interested only in
the subset of context-dependent follow-up questions, i.e., that require some additional information from the dialogue context in order to be fully specified and unambiguous. In fact, the Hamburg focus tree includes dedicated focus nodes that specifically cover context-dependent follow-up questions. These focus nodes were assigned a special “context-dependent” attribute by Hamburg’s domain experts, which is interpreted by the focus tree search algorithm in that it searches these nodes first, and only in the specific dialogue context for which the domain experts have foreseen the follow-up question. What follows is an example taken from the Hamburg focus tree (and re-used for BoB). After the user has asked about the availability of guided tours to the library, he asks an elliptical follow-up question: “Where is the meeting point?”. This follow-up matches with the regular expression pattern from a focus node marked specifically as context-dependent, whose pattern only requires the presence of “where” in the question.

Re-using subdialogues

Subdialogues are used to encode relatively short, predefined sequences of system questions, to which the user’s answer must come from a small, predefined set of possible answers (e.g., “yes/no” for simple questions). From the domain expert’s point of view, they allow users to be guided through the domain by pointing them to relevant options in specific dialogue situations. This kind of guidance should be especially useful for inexperienced users who do not know how to formulate their problem or question explicitly. Subdialogues must be handled explicitly by the search algorithm, since their regular expression patterns must only be searched when the corresponding subdialogue is active. This prevents focus nodes with unspecific patterns like “yes” to be selected in the course of the global focus tree search (steps 3 and 4 of the algorithm described at the end of this section).

The decision in favor of importing the pre-defined subdialogues comes at a cost. The way that subdialogues are included in the focus tree using a special type of focus nodes breaks the otherwise clean and purely declarative nature of the tree structure. Besides encoding topics and sub-topics, the tree now contains nodes with procedural semantics, which require the focus tree search algorithm to follow a hard-coded link to some other (possibly remote) node, where the next user turn can then be processed as a continuation of the subdialogue. One solution for separating the declarative topic hierarchy from these procedural additions would be to have a more powerful dialogue manager that could generate subdialogue sequences on the fly, given information about the current topic. It would be a possible step towards a better understanding and control of system initiative in an IQA system.

---

5In the surface form, the follow-up question is lacking the attribute “…for guided tours?”. 
Searching the focus tree

We now describe the current implementation of BoB’s search algorithm, and how it works in conjunction with the focus tree introduced earlier in this section. Every time the user enters a new question, a suitable node in the focus tree has to be identified, so that the system response stored in that specific topic node can be returned as an answer. The search for a focus node depends both on the user input and on the previously active focus node. By starting each search for the next system response at the currently active focus node, we take advantage of the underlying notion of the focus tree, that topics which describe likely continuations of the conversation are close to each other in terms of node distance. Figure 1.1 shows a flow chart of the focus tree search algorithm as it is currently implemented in BoB. For compactness of the diagram, the following notation is used:

<table>
<thead>
<tr>
<th></th>
<th>the Current focus node</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD</td>
<td>a SubDialogue focus node</td>
</tr>
<tr>
<td>CD</td>
<td>a Context-Dependent focus node</td>
</tr>
<tr>
<td>N</td>
<td>a “normal” focus node (i.e., neither SD nor CD)</td>
</tr>
<tr>
<td>Match</td>
<td>specific focus node retrieved by previous search operation</td>
</tr>
<tr>
<td>.link</td>
<td>link attribute of SD, pointing to specific focus node where subdialogue processing will resume</td>
</tr>
<tr>
<td>.sysResponse</td>
<td>system response encoded in focus node</td>
</tr>
<tr>
<td>SD-Mode</td>
<td>flag indicating if system is currently in a subdialogue</td>
</tr>
<tr>
<td>“matching”</td>
<td>focus node’s regular expression pattern matches current user input</td>
</tr>
</tbody>
</table>

Conceptually, the search algorithm can be divided into two parts. In the first part (consisting of the pattern matching steps marked with (1) and (2) in the diagram), subdialogues and context-dependent follow-up questions are dealt with on a local level, i.e., without the current focus shifting to a node more remote than the siblings nodes. In the second part of the algorithm (pattern matching steps (3) and (4)), the search for a system response is iteratively extended to the entire focus tree.

5 Conclusion and future directions

Currently, the contents and the topology of the focus tree are determined entirely by domain experts based on their intuition. We are currently exploring systematic ways of constructing or extending a focus tree, so as to get an understanding of which (follow-up) user questions will be covered by the system (manuscript in preparation). What we clearly lack at this point is an evaluation of the BoB system with respect to real user data. We are considering different possibilities for this. Regarding BoB’s coverage of the most
Applying a Focus Tree Model of Dialogue Context to Interactive Question Answering

Figure 1.1: The BoB focus tree search algorithm
important topics of user questions, we have only a preliminary result based on the not yet adjusted Hamburg focus tree (cf. Kirschner (2006)). A more thorough user study can be conducted once the focus tree has been completely adjusted by our domain experts (see below). This user study will also have to verify the ability of the system to handle context-dependent follow-up questions via the special “CD” focus nodes described earlier. We are currently studying how well our focus tree-based approach is able to model (context-dependent) follow-up user questions, using our previously collected corpus of Wizard-of-Oz dialogues, and how this depends on the topology of the focus tree (manuscript in preparation).

Although the current BoB system is simplistic (e.g., lacking linguistic knowledge, and an explicit representation of natural language semantics or the pragmatics of dialogue), the advantages of our approach are clear: we were able to build a working IQA system from scratch in a relatively short time (around 1 year). At the time of writing, a team of domain experts is working on the localization of the focus tree in terms of the covered topics and the two additional target languages, using tools described in Kirschner (2007). Once this task is finished, we will be able to compare different ways of modeling dialogue context in IQA (based on focus trees), using the running BoB system as a test bed. Given dialogue log files, we plan to study patterns of topic change within IQA dialogues.

Acknowledgements

We would like to thank Bonnie Webber for the discussions about IQA; her talk on that topic, delivered at the Free University of Bozen-Bolzano in November 2006, inspired especially Section 2.

Bibliography


Applying a Focus Tree Model of Dialogue Context to Interactive Question Answering


Manuel Kirschner

Total Lexicalism in Language Technology

JUDIT KLEIBER
Department of Linguistics, University of Pécs, Hungary
kleiberj@btk.pte.hu

Abstract. The paper summarizes a four-year project, whose aim has been to prove that total lexicalism is worth applying to computational linguistic tasks. Total lexicalism means that all the information (needed for a sentence to be put together) is stored in the lexicon, thus there is no need for language-specific syntactic rules. A Prolog-implementation has been made on this basis, which can (on a small corpus) decide whether a sentence is grammatical, and can provide morphological, syntactic and semantic representations. The lexical items of the parser are Hungarian and English stems and affixes. By means of two-way application of the program (parsing and generating), machine translation is also achieved.

1 Introduction

In the last decades, lexicalism became an important issue in generative linguistics. It has always been admitted that a grammar needs a lexicon, where the words can be found with some of their properties. Later the importance of this lexicon has increased: more and more features became part of the lexicon, thus less and less rules the syntactic component had to contain.

(Karttunen 1986) introduces radical lexicalism by using a unificational categorial grammar. In this grammar the only syntactic operation is function application, most of the information is stored in the lexicon, and grammaticality of sentences can be decided by means of unification. This grammar is especially suitable for phenomena like nonlocal dependencies and languages with free word order. Phrase structure grammars usually have difficulties with both of them.

(Alberti 1999) takes the idea even further, and defines a totally lexicalist grammar (GASG), which is a modified unificational categorial grammar. From this grammar even function application is omitted, thus unification remains the only operation. This yields to a lexicon richer than any earlier one, where all the information is stored in descriptions of lexical items.¹

¹This intention coincides with two mottoes of Joshi’s (Joshi 2003): “Complicate Locally, Simplify Globally”, and “Grammar ≈ Lexicon”
The aim of our research team has been to try out whether total lexicalism is worth applying to computational linguistic tasks. (Karttunen 1986) proved that using a unificational categorial grammar (in theory) is very efficient in the case of agglutinative languages. This could predict that an implementation of a totally lexicalist grammar would work well in the case of Hungarian. However, we had to prove that the idea can be applied to completely different languages as well; this is why we added English lexical items to the database.

We have made the implementation in Prolog, which is suitable for computational linguistic tasks if the database does not need to be large. Hence our goal has been to try out lexicalist methods, and not to produce a software; we did not need a huge lexicon, only a few hundred entries. Our program can parse Hungarian and English sentences, and can provide morphological, syntactic and semantic representations. We have also elaborated a totally lexicalist approach to machine translation, which is achieved by the two-way application of the program (parsing and generating).

Section 2 tells about lexicalism in language technology nowadays, how successful unification-based parsers are. Section 3 introduces our starting point, why we considered that this field is worth doing research into. Section 4 expresses what our goals and expectations have been. Our achievements can be read in Section 5, with examples from the program. Section 6 is about the limitations this approach might have, and the possible solutions to these problems, including the directions of further work. Finally to conclude, the significance of our results is explained in Section 7: why we find total lexicalism suitable for computational linguistic applications.

2 Lexicalism in language technology

Lexicalism proved to be successful not only in theory but in the field of language technology as well. Parsers based on lexicalist grammars can provide more detailed analysis than those based on phrase-structure grammars (or than parsers not using deep linguistic methods), they can handle languages with rich morphology and free word-order as well, and the outputs of these analyses can be parallel, thus machine translation can be achieved more easily.

Previously existed parsers did not turn out to be sufficient enough for intelligent applications such as question answering, text summarization, or good-quality machine translation. Deep-linguistic methods seemed to be indispensable for completing tasks like these. Phrase-structure grammars usually have difficulties with languages like Hungarian (rich morphology, almost free word-order), and they sometimes have too complicated and very different rule-systems for various languages. Lexicalist approaches seem to avoid these problems. Coverage has been a secondary issue (many of these
applications are still in experimental phase), but some of these parsers has actually reached the coverage of parsers using shallow techniques and statistical methods.

A further advantage of rule-based approaches (as opposed to example-based ones) is their re-usability: e.g. systems developed for parsing can be applied to question-answering or machine translation. A system can only be suitable for intelligent applications if a semantic representation is assigned to a sentence: which some unification-based programs can accomplish.

Two of these systems are certainly worth mentioning: the Parallel Grammar project (Butt, King, Masuichi, and Rohrer 2002), which uses LFG formalisms; and the English Resource Grammar (ERG), which is the largest HPSG-based grammar for English, implemented in the LKB (Linguistic Knowledge Building) platform (Copestake 2002). In the case of HPSG, even a “starter-kit” has been developed to make grammar-writing easier (the Grammar Matrix project, (Bender, Flickinger, and Oepen 2002)). Grammars are implemented for several languages within both projects (e.g. English, German, Japanese, Norwegian, and Urdu); some of them also contain a semantic component (using MRS, Minimal Recursion Semantics).

Machine translation is aimed within these project as well: partly to prove the universality of their formalism, and partly for practical reasons: to create good-quality translations, which has not really been an issue earlier. The results are promising, though most of these systems have a rather small database (so far).

3 Total lexicalism – starting point

The success of lexicalist approaches encourage us to keep trying out the “extreme” possibility of total lexicalism: can a grammar be developed (in theory and in practice) if only lexicon exists, syntax (PS trees) does not? (Schneider 2005) raises the idea of reducing c(onstituent)-structure from LFG-representations as much as possible in order to make the parsing simpler (this seems to be necessary especially in the case of languages with free word order). In our proposal c-structure is fully eliminated, and only an f-structure-like representation is given.

The aim of our research team has been to prove that total lexicalism is worth applying in language technology. A totally lexicalist grammar does not build phrase structure trees, and does not define rules. It has a huge lexicon, which contains lexical items with all their properties, and uses unification. The arguments for applying a grammar like this are as follows.

(1) Since phrase structure grammars usually have difficulties with several phenomena, such as nonlocal dependencies and free word order, applications which are based on them and do not use any other method, probably cannot reach one hundred percent accuracy.
(2) Grammars of various languages can be highly different, thus a system using only phrase structures can be suitable for one particular language (Mitkov 2003) (or only few very similar ones). This also makes machine translation difficult to achieve, since separate mechanisms and rule-structures are needed for every language pair. On the contrary, a totally lexicalist grammar is not language-specific.

(3) Rule-based systems may not be accurate enough, because of exceptions which every language has. In this respect lexicalist approaches are much more promising.

(4) A monostratal grammar can be more effective at the parsing process. The unification of two elements takes more time and effort, but false solutions are excluded sooner. GASG is not only monostratal but homogeneous as well. One of our goals is to try out whether this property increases or decreases efficiency.

It is for these reasons that we have decided to try out totally lexicalist methods in computational linguistics. The background grammar is GASG (Generative/Generalized Argument Structure Grammar, (Alberti 1999)). We have made some changes though, when it seemed rational during the implementation. An important difference is that in GASG lexical items are inflected words, but in the case of Hungarian it seemed much more effective to store morphemes in the lexicon, and have a morphological component.

In GASG descriptions of lexical items have four components: the own features of the element (e.g. its part of speech category), its requirements in a sentence (properties of possible arguments), semantic description (a proto-DRS), and the connection between syntax and semantics. A sentence is grammatical if all the requirements of the given lexical items are met, that is, when unification is successful.

Without building phrase-structure trees GASG could be regarded as a dependency grammar, which is not effective computationally: it is proved that without restrictions to word order, the parsing algorithm is exponential. But this does not stand in the case of GASG, because there is a special requirement which is in charge of word order, namely rank parameters. We assume that every word wants to be next to every other word if they are in a semantic relation in a sentence. Obviously, all these requirements cannot be met; this is why they should be ranked (the strongest wins). This system can easily explain word order differences between languages as well: ranks are different. For example, in Hungarian free adverbs can appear before, after or inbetween the arguments, while in English only at the beginning or at the end of a sentence. The explanation is that in Hungarian the ranks of arguments and free adverbs are equal, but in English the rank of a free adverb is weaker then the rank of any of the arguments.

\[\text{Of course in the case of very similar languages, shallow parsing can be enough (Homola and Kubon 2004). Total lexicalism pays back when the languages are quite different.}\]
4 Expectations

Our starting aim has been to prove that GASG is an exact, strictly formalized grammar, and we assumed that a working implementation could be the best evidence for that. While implementing the grammar in Prolog, we found that totally lexicalist methods can be very useful in language technology, especially in the case of languages like Hungarian, which is an agglutinative language with (almost) free word order. But we have also added English lexical items to the database to prove that the methods work for other kinds of languages as well.

Our practical aim has been to make a parser, which can decide whether a sentence is grammatical or not, and (in the case of grammatical sentences) can print out various representations: morphological (the relevant lexical items), syntactic (relations among them) and semantic (a DRT). Details on these representations can be read in section 4.

The most important component is semantics. This is not just because few computational systems contain semantic representations (mainly lexicalist parsers), but because this semantic output can be regarded as a machine-aided translation. Learning to read these DRT-like representations (which are in English) takes only a few hours, while learning e.g. Hungarian takes years.

Finally, we aimed to work out a mechanism for machine translation based on total lexicalism. We intended to use the built-in generating function of Prolog for this task. We assumed that this can be done through semantics by means of the two-way application of our program: parsing source language sentences, then generating target language sentences (which includes their parsing as well – checking grammaticality).

So our main purpose has been to prove that GASG is an exact, strictly formalized grammar, and that totally lexicalist methods are worth applying in language technology. We intended to accomplish this task by making an implementation in Prolog. We chose using smaller database but adding various kinds of lexical items in Hungarian and English. Making a huge lexicon and so a marketable software has not been one of our goals so far, enlarging the size of the database could be our next step.

5 Achievements

The present parser is in Prolog, and can carry out three tasks. It (1) decides (on a small Hungarian and English corpus) whether a sentence is grammatical or not, (2) assigns various types of representations to grammatical sentences, and (3) translates these sentences from Hungarian into English and vice versa.

The input of the parser is a series of words. The program first checks
whether the words are well-formed. To accomplish this task, it has to segment the words into morphemes (lexical items) and check all the (morpho)phonological requirements these elements have. If unification is successful, the list of the relevant lexical items is printed out.

In Hungarian, words (especially suffixes) can appear in several surface forms, so (morpho)phonological requirements can be very complicated. For example, the accusative suffix has five allomorphs: -t, -ot, -at, -et, and -t. It depends on several factors: the sound right before the suffix, the frontness of the stem, the roundness of the previous morpheme, and a so-called lowering property (which cannot be calculated on the basis of the phonological form of the morpheme). This is why the own word of a lexical item (how it appears in a sentence) often contains variables.

Let us see a simple example, first in Hungarian. If the grammaticality of (1) is asked, the parser finds it correct, and prints out (2), the list of the relevant lexical items.

(1) P´eter énekel-tet-het-i Mari-t.
Peter sing-CAUSE-MAY-3SG Mary-ACC
‘Peter may make Mary sing.’

(2) LEXICAL ITEMS:
Péter: n(1,1,li(m("","Péter",""),labstem("Peter",phonfst...,1,[])))
énkel: n(2,1,li(m("","énkel",""),labstem("sing",phonfst...)))
tet: n(2,2,li(m("t","A","t"),labder("cause",phonfsu(2,2,0,2,2),2,...)))
het: n(2,3,li(m("h","A","t"),labsuff("may",phonfsu(1,1,2,1,2))))
i: n(2,4,li(m("","i",""),labsuff("sg3obj+def",phonfsu(1,3,1,3,2,3))))
Mari: n(3,1,li(m("","Mari",""),labstem("Mary",phonfst(2,2,0,2,1,[])))
t: n(3,2,li(m("V","t",""),labsuff("ACC",phonfsu(1,1,3,1,4))))

Each lexical item gets a numbering in the sentence, which makes parsing simpler. For example, in the case of the allomorph ‘het’ it is (2,3), which means that ‘het’ is the third morpheme of the second word. After this numbering the own word can be seen with variables (capital letters), which is divided into three parts for technical reasons.

Finally a label can be seen, which is different in the case of stems (labstem), derivative elements (labder), and other kinds of suffixes (labsuff). The reason for that is the difference between the relevant properties we need to store. For instance, with stems the important phonological features are the frontness and roundness of the item, and whether it is a lowering stem or not. In the case of suffixes, the question is whether the suffix causes various stem alternations: lengthening, shortening, vowel-zero alternation or lowering (1 stands for yes, 2 stands for no, 3 stands for irrelevant). Besides, we need to store part-of-speech categories (with each type), argument-structures (with stems and derivative element), and rank parameters (with suffixes for the right morpheme order within words).

The next step is syntactic analysis. I do not put the output of the parser
in the paper because it is quite complicated, only a figure which shows the relations among the lexical items.

![Syntactic relations](image)

**Figure 1.1: Syntactic relations**

Arrows with two heads show the two-way relations (predicate-argument pairs), arrows with one head represent one-way (free) relations. Predicates search for their arguments in two pillars: a nominative and a determining one. In this example arguments are proper names, so these two pillars coincide in the case of the subject. The object of the sentence is found in the accusative suffix -t, and the causative -tAt needs it in the sentence, because the verb stem is intransitive.

Finally, a semantic representation is printed out. It is a kind of DRS (Discourse Representation Structure, (van Eijck and Kamp 1997)), but it has additional condition rows as well, which are responsible for building the actual DRS into wider context (LDRS, (Alberti 2000)).

\[
\text{(4) SEMANTICS:}
\begin{align*}
\text{provref} & ("fixpoint", [e(2,3,1)]) \\
\text{provref} & ("old", [x(1,1,1)]) \\
\text{pred} & ("Peter", 1, [x(1,1,1)]) \\
\text{provref} & ("new", [e(2,1,1)]) \\
\text{pred} & ("sing", 2, [e(2,1,1), x(3,1,1)]) \\
\text{provref} & ("new", [e(2,2,1)]) \\
\text{provref} & (>\text{"="}, [e(2,2,1), e(2,1,1)]) \\
\text{pred} & ("cause", 2, [e(2,2,1), x(1,1,1), e(2,1,1)]) \\
\text{provref} & ("new", [e(2,3,1)]) \\
\text{provref} & (>\text{"<"}, [e(2,3,1), e(2,2,1)]) \\
\text{pred} & ("may", 2, [e(2,3,1), e(2,2,1)]) \\
\text{provref} & ("old", [x(3,1,1)]) \\
\text{pred} & ("Mary", 3, [x(3,1,1)])
\end{align*}
\]

**Figure 1.2: Semantics**

When the grammaticality of the English version of this sentence is asked, very similar representations are printed out, only the “names” of the referents (which are given on the basis of the numberings of the morphemes) are different.
Our program uses this semantic representation for translation, and another representation (so-called copredication network), a level between syntax and semantics. Because of the near-universality of semantics, translating from and into completely different languages (like English and Hungarian) is not more difficult than it would be in the case of similar languages. (5) shows the results of translating (1) into English, and (6) is the translation of the English version of (1) into Hungarian.

(5) translate\textsubscript{Hun,Eng}("Péter énekeltetheti Marit.").
In English: Peter may make Mary sing.
yes

(6) translate\textsubscript{Eng,Hun}("Peter may make Mary sing.").
In Hungarian: Péter énekeltetheti Marit.
yes

Because the program stores lexical items, not words, it is not problematic that in the Hungarian version the sentence consists of three words, while the English version contains five.

A more extreme case can be when the subject and the object are not present in the Hungarian sentence either, which is possible, since Hungarian is a pro-drop language. In this case verbal suffixes show the person and the number of the missing elements. This could be even harder to a translator using traditional lexicons. But our parser can assign a semantic representation to a sentence like this as well, so translating it would not be more difficult than translating sentences with spelled-out arguments. In (8) the translation of the simple sentence (7) can be seen, together with the morphological, syntactic and semantic representations.

(7) Szeret-l-ek.
love-2SG.OBJ-1SG.SUBJ
‘I love you.’

(8) translate\textsubscript{Hun,Eng,print}("Szeretlek.").
LEXICAL ITEMS:
szeret: n(1,1,li(m("","szeret",""),labstem("love",phonfst(1,2,2),2,[["NOM","ACC"]])))
l: n(1,2,li(m("","l",""),labsuff("objperson2",phonfsu(3,2,1,1),2,2.5)))
ek: n(1,3,li(m("V","k",""),labsuff("sg1",phonfsu(1,1,2,3),2,3)))
SYNTAX:
gr("suff","stem","free",1,2,1,1)
gr("suff","stem","free",1,3,1,1)
SEMANTICS:
provref("fixpoint",[e(1,1,1)])
provref("new",[e(1,1,1)])
pred("love",1,[e(1,1,1),r(0,1,1),r(0,1,2)])^9

^9r011 means I: a built-in referent (0 shows that) singular (1) and first person (2). r012 means you: built-in referent, singular, second person.
In English: I love you.

So we have made a parser for Hungarian on the basis of GASG, a totally lexicalist grammar. We also added English lexical items to prove that the mechanisms work not only for this particular language. Our program can decide (on a small corpus) whether a sentence is grammatical or not, and can produce various types of representations, among which the most important is semantics. Using this and the generating function of Prolog, we can also translate from Hungarian into English and vice versa.

6 Limitations

While making the implementation in Prolog we came across several difficulties. For instance, sometimes the database needed to be modified, when we found extra properties, which should be stored. This could not be an easy task in Prolog. Another disadvantage of this programming language is that its output cannot be easily read.

This is why we decided to rebuild the lexicon as a relational (SQL) database, and so our lexicon has become compatible with the XML-format as well. Because of the new structure, our lexicon has been able to be used in other fields, too. For instance, it can be regarded as a “dynamic corpus”. The expression means that this lexicon would not contain the existing words (sentences), like a regular corpus, but the possible ones which could be generated. Furthermore, the users could look up not only words (sentences) but elements with particular features as well (e.g. Hungarian lowering stems). Another application could be helping education: teaching foreign language (Hungarian) or grammar.

Another disadvantage of this approach could be the fact, that it is competence-based. More and more linguists think that corpus-based approaches are more promising; or – if a system is competence-based – the rules should be more flexible to be able to handle a text with mistakes as well (Prószyński 2005).

The solution to this problem could be using a special feature which we have already tried in the new system (SQL database). Because of the locality of our approach (“rules” are assigned to lexical items, not the whole language), the grammar is flexible. We can easily “switch off” any property at any lexical item, so that the set of grammatical sentences would be just a little different.

Finally, the idea of total lexicalism may have a disadvantage as well. Treating general syntactic rules could be problematic, e.g. the rule in English that every sentence has a subject. The question is where to store this feature. Putting it into the description of every verb stem would not be very effective. The solution to this problem can be to find one particular morpheme which...
the feature should be assigned to.

Considering these difficulties we would like to make some changes in the future. We plan to improve the semantic component on the basis of (Alberti 2005), to be able to handle texts, not only sentences, and to get a more detailed analysis, including a semantic representation more sophisticated than any earlier one. Our ultimate goal is to achieve good-quality machine translation which would also account for rhetorical relations, discourse-functions (topic, focus), or aspect. We believe that this semantic representation is detailed enough to serve as an interlingua, which could make it easier to achieve language-independent machine translation. (Lexicalist approaches like LFG and HPSG usually use transfer-based machine translation, which needs different transfer lexicons for every language pair.)

Furthermore, we plan to switch to a more effective programming language, and enlarge the size of the database. Meanwhile, we would like to achieve goals which do not need a large corpus, such as helping education. And finally, we plan to keep working on elaborating various phenomena (derivative system of Hungarian, argument structures, etc.) to prove that total lexicalism can be an effective tool in language technology.

7 Conclusion

The aim of our research team has been to prove that a totally lexicalist grammar can be a wellworking system in theory and in practice alike. To achieve this goal, we have made a parser in Prolog which can decide the grammaticality of a sentence and can provide morphological, syntactic and semantic representations. Our small database consists of Hungarian and English stems and affixes, and can also translate from Hungarian into English and vice versa.

In the past four years we tried out several linguistic ideas. We experimented with phenomena people usually do not do research into. We have equally studied details (e.g. the behavior of Hungarian articles), and larger issues (e.g. translation). We could afford to do so because our aim has not been to produce a marketable software as soon as possible. Our parser obviously needs to be extended, but our results are promising. The significance of our project is that we have showed that morphology-based total lexicalism and representational discourse semantics are worth applying in language technology. We intend to strengthen this view by further research in the future.

Acknowledgements

I am grateful to the Hungarian National Scientific Research Fund (OTKA K60595) for their contribution to my costs.
Bibliography


A Multi-Modal Combinatory Categorial Grammar analysis of Japanese Nonconstituent Clefting

YUSUKE KUBOTA†
The Ohio State University
kubota@ling.osu.edu

E. ALLYN SMITH†
The Ohio State University
esmith@ling.osu.edu

ABSTRACT. Despite the notion that clefting is a cross-linguistic constituency test, Japanese allows some nonconstituent exceptions. There is, however, a certain restriction on the degree of flexibility; some constituents are more tightly connected (and thus less likely to be separated by clefting) than others. We refine Kubota and Smith's (2006) CCG account in terms of Multi-Modal CCG (Baldridge, 2002): finer-grained modal control provides a means for capturing different degrees of connectedness between an argument and its functor. We then demonstrate how a MMCCG system that finds independent motivation from syntactic complex predicate data interacts with a simple analysis of clefting to account for the full range of clefting patterns. This in turn suggests that what seems to pose problems for a simple analysis of a given phenomenon (clefting) can be overcome once interactions with other phenomena are taken into account.

1 Introduction

Multi-Modal Combinatory Categorial Grammar (MMCCG) (Baldridge, 2002) brought together two strands of research in categorial grammar: Combinatory Categorial Grammar (CCG), a more linguistically-oriented variant that has entertained a wider range of empirical applications and Type-Logical Grammar (TLG), a more formally-oriented variant whose logical properties are better understood. Accounting for natural language data, both cross-linguistically and across phenomena within a single language, is crucial for developing a formal theory of natural language. While crosslinguistic work already exists for MMCCG (Baldridge, 2002), the in-depth description of

†The authorship of this paper is fully joint; the authors are listed alphabetically.
a single language has yet to be undertaken, to the best of our knowledge (and is unfortunately very rare in categorial grammar as compared to other grammatical theories). Thus, this paper takes a first step in that kind of investigation by giving a detailed analysis of the cleft construction in Japanese within a larger theory of grammar of Japanese that handles scrambling and complex predicates. We believe that this kind of work represents a truly interdisciplinary study of logic and language, wherein implications of empirical data are seriously taken into account in theory development through the process of modelling complicated interactions of linguistic phenomena explicitly within a formal theory.

2 Data

In Japanese, a cleft sentence is formed by topicalizing a sentence (marked by the topicalizer *wa*, which in turn requires the nominalizer *no*) and combining it with the focused element (an argument or an adjunct missing from the topicalized sentence) followed by the copula *da*.\(^1\)

\[(1) \ [ \text{Ken ga} \ t_i \, \text{kat-ta}] \, \text{no} \, \text{wa} \, \text{sono hon} \, (o) ; \text{da.} \]

\[\text{Ken NOM \ buy-PAST NMLZ TOP that book ACC COP} \]

\[\text{‘It is that book that Ken bought.’} \]

In (1) the object *sono hon* is missing, appearing instead in the position immediately preceding the copula. Just as in other languages, these cleft sentences are truth-conditionally equivalent to simple sentences but differ in their information structure depending on what is clefted.

In addition to simple constituent clefts such as those in (1), Japanese allows nonconstituent clefts as in (2), as was first noted by Koizumi (1995).\(^2\)

\[(2) \ a. \ [ \text{Ken ga} \ t_i \, t_j \, \text{barasi-te simat-ta}] \, \text{no} \, \text{wa} \, \text{Mari ni} \]

\[\text{Ken NOM \ disclose EMPH-PAST NMLZ TOP Mari DAT} \]

\[\text{sono himitu o_j} \, \text{da.} \]

\[\text{that secret ACC COP} \]

lit. ‘It is to Mari the secret that Ken (inadvertently) disclosed.’

\[b. \, [t_i \, t_j \, t_k \, \text{Barasi-te simat-ta}] \, \text{no} \, \text{wa} \, \text{Ken ga} \_ \, \text{Mari ni_j} \]

\[\text{disclose EMPH-PAST NMLZ TOP Ken NOM Mari DAT} \]

\[\text{sono himitu o_k} \, \text{da.} \]

\[\text{that secret ACC COP} \]

lit. ‘It is Ken the secret to Mari that (inadvertently) disclosed.’

\(^1\)The use of italics in examples indicates the focal position, while the use of brackets indicates the topic position (the focus/topic division roughly corresponds to new/old information distinction); brackets and traces appear for expository ease.

\(^2\)It should be noted that there are speakers who do not accept sentences of this sort (Kizu, 2005). The judgements of the sentences reported here are those of the native-speaking author of this paper.
In these examples, multiple arguments have been clefted together. It is also possible to have multiple adjuncts or argument/adjunct pairs in the focal position. For relevant data, see Kubota and Smith (2006).

Furthermore, the order of elements in the focal position is flexible (that is, if the orders of the accusative and dative objects are switched in (2a), that will still yield a grammatical sentence). This is presumably related to the fact that word order is relatively free in Japanese. That is, Japanese is a verb-final language but allows for scrambling of arguments of the verb. Thus, both sentences with SOV and OSV orders are grammatical.

But there are also some limitations on clefting. (3a) is a case involving a complex predicate construction with the -te morphological marking on the embedded verb.\(^3\) Essentially, the ungrammaticality of this example is due to the fact that the embedded verb (yon-de ‘read’) and the matrix verb (morat-ta (benefactive)) are separated from one another. Similarly, adjectives modifying nouns cannot be clefted because they cannot be split from those nouns, as in (3b).

(3) a. *[Morat-ta] no wa Ken ga Mari ni sono hon
   BENEF-PAST NMLZ TOP Ken NOM Mari DAT that book
   o yon-de da.
   ACC read-MKR COP
   lit. ‘The thing that was done for the benefit of somebody was
   that Ken had Mari read that book for him.’

   b. *[t\(\_\)Hon o Taro ga yon-da] no wa nagai da.
   book ACC Taro NOM read-PAST NMLZ TOP long COP
   intended: lit. ‘It is long that Taro read a book.’

3 Kubota and Smith’s (2006) analysis of nonconstituent clefting

We now review the previous analysis of Japanese nonconstituent clefting by Kubota and Smith (2006) (K&S) in CCG. Essentially, in K&S’s analysis, argument clusters that appear in the focal position of sentences like those in (2) are treated as constituents, employing the technique familiar from the treatment of nonconstituent coordination (Dowty, 1988; Steedman, 1996).

K&S make use of basic CCG combinatory rules of Function Application (FA), Type-Raising (TR) and Function Composition (FC).
4 For lexical entries, K&S assume that each verb has a separate entry for each possible order in which it takes its arguments (thus, a ditransitive verb is assigned

\(^3\)We call this construction the ‘-te form complex predicate’. In (3a), the morpheme appears in the allomorph -de with voicing on the initial consonant.

\(^4\)K&S use the Lambek style slash notation. We depart from this and adopt the ‘result leftmost’ notation that is more commonly adopted in the literature of CCG.
A Multi-Modal Combinatory Categorial Grammar analysis of Japanese Nonconstituent Clefting

eight separate lexical entries). In addition to this assumption, the following lexical entries for function words are posited:

\[
\begin{align*}
(4) & \quad \text{a. no: } (S[+\gamma]\$)\backslash(S[+\gamma]\$) \\
& \quad \text{b. wa: } (S[+\gamma]\$)\backslash(S[+\gamma]\$) \\
& \quad \text{c. da: } (S[-T]\backslash X)\backslash(S[+T]/X)
\end{align*}
\]

The nominalizer no and the topic marker wa are identify functions over S-rooted categories (using the $ convention of CCG (Steedman, 2000)). The features N (nominalized) and T (topicalized) are binary features governing the distribution of these function words (i.e., the feature specifications of these words ensure that a sentence without no and wa is an ungrammatical Japanese cleft sentence). The sentence-final copula da plays a pivotal role in putting together the topicalized and focused elements by changing the directionality in which the focused element looks for its argument.

K&K’S derivation for the argument-cluster cleft sentence in (5) appears in (6), where the different parts of the derivation are split for readability, with the third piece showing how the first two ultimately combine:

\[
\begin{align*}
(5) & \quad \text{[Ken ga watasi-ta] no wa sono hon o Mari ni da.}
& \quad \text{Ken NOM give-PAST NMLZ TOP that book ACC Mari DAT COP lit. ‘It is that book to Mari that Ken gave.’}
\end{align*}
\]

\[
\begin{align*}
(6) & \quad \text{Ken ga watasi-ta no wa sono hon o Mari ni da.}
& \quad \text{Ken NOM give-PAST NMLZ TOP that book ACC Mari DAT COP lit. ‘It is that book to Mari that Ken gave.’}
\end{align*}
\]

The first part of the derivation shows how the topicalized sentence is formed: FA combines the verb (the entry yielding the OSV order in simple sentences) with the subject, where the object remains unsaturated.\(^5\) In the focal position, both of the missing arguments are first type-raised and the resultant categories are combined via FC, yielding a functor that is looking for the rest of the sentence (namely, the topicalized portion) to become a complete sentence. However, it is looking for this argument in the wrong direction, namely, to the right. The copula crucially comes into play here and flips the directionality of the slash, causing it to look to the left, as seen at the last step of the second part.

\[^5\text{The derivation in (6) makes use of FC in combining no and wa, but there is a fully equivalent derivation involving only FA as well.}\]
At this point, we have seen how K&S’s system handles nonconstituent clefting via the interaction of TR and FC. But there are a few points that remain unsatisfactory in this account. Given the flexibility of CCG, that analysis predicts that any string of words that can occur on the leftmost edge of a sentence can be clefted. Thus, it overgenerates sentences such as those in (3) from section 2. The following examples clarify the pattern we find in the complex predicate data:

7 hon o

\[ \begin{array}{c}
\text{TR} \\
\text{NP}_a \\
\text{NP}_b \\
\text{NP}_c \\
\text{NP}_d \\
\text{NP}_e \\
\text{NP}_f \\
\text{NP}_g \\
\text{NP}_h \\
\text{NP}_i \\
\text{NP}_j \\
\text{NP}_k \\
\text{NP}_l \\
\text{NP}_m \\
\text{NP}_n \\
\text{NP}_o \\
\text{NP}_p \\
\text{NP}_q \\
\text{NP}_r \\
\text{NP}_s \\
\text{NP}_t \\
\text{NP}_u \\
\text{NP}_v \\
\text{NP}_w \\
\text{NP}_x \\
\text{NP}_y \\
\text{NP}_z \\
\text{NP}_A \\
\text{NP}_B \\
\text{NP}_C \\
\text{NP}_D \\
\text{NP}_E \\
\text{NP}_F \\
\text{NP}_G \\
\text{NP}_H \\
\text{NP}_I \\
\text{NP}_J \\
\text{NP}_K \\
\text{NP}_L \\
\text{NP}_M \\
\text{NP}_N \\
\text{NP}_O \\
\text{NP}_P \\
\text{NP}_Q \\
\text{NP}_R \\
\text{NP}_S \\
\text{NP}_T \\
\text{NP}_U \\
\text{NP}_V \\
\text{NP}_W \\
\text{NP}_X \\
\text{NP}_Y \\
\text{NP}_Z \\
\text{NP}_a \\
\text{NP}_b \\
\text{NP}_c \\
\text{NP}_d \\
\text{NP}_e \\
\text{NP}_f \\
\text{NP}_g \\
\text{NP}_h \\
\text{NP}_i \\
\text{NP}_j \\
\text{NP}_k \\
\text{NP}_l \\
\text{NP}_m \\
\text{NP}_n \\
\text{NP}_o \\
\text{NP}_p \\
\text{NP}_q \\
\text{NP}_r \\
\text{NP}_s \\
\text{NP}_t \\
\text{NP}_u \\
\text{NP}_v \\
\text{NP}_w \\
\text{NP}_x \\
\text{NP}_y \\
\text{NP}_z \\
\text{NP}_A \\
\text{NP}_B \\
\text{NP}_C \\
\text{NP}_D \\
\text{NP}_E \\
\text{NP}_F \\
\text{NP}_G \\
\text{NP}_H \\
\text{NP}_I \\
\text{NP}_J \\
\text{NP}_K \\
\text{NP}_L \\
\text{NP}_M \\
\text{NP}_N \\
\text{NP}_O \\
\text{NP}_P \\
\text{NP}_Q \\
\text{NP}_R \\
\text{NP}_S \\
\text{NP}_T \\
\text{NP}_U \\
\text{NP}_V \\
\text{NP}_W \\
\text{NP}_X \\
\text{NP}_Y \\
\text{NP}_Z \\
\text{NP}_a \\
\text{NP}_b \\
\text{NP}_c \\
\text{NP}_d \\
\text{NP}_e \\
\text{NP}_f \\
\text{NP}_g \\
\text{NP}_h \\
\text{NP}_i \\
\text{NP}_j \\
\text{NP}_k \\
\text{NP}_l \\
\text{NP}_m \\
\text{NP}_n \\
\text{NP}_o \\
\text{NP}_p \\
\text{NP}_q \\
\text{NP}_r \\
\text{NP}_s \\
\text{NP}_t \\
\text{NP}_u \\
\text{NP}_v \\
\text{NP}_w \\
\text{NP}_x \\
\text{NP}_y \\
\text{NP}_z \\
\text{NP}_A \\
\text{NP}_B \\
\text{NP}_C \\
\text{NP}_D \\
\text{NP}_E \\
\text{NP}_F \\
\text{NP}_G \\
\text{NP}_H \\
\text{NP}_I \\
\text{NP}_J \\
\text{NP}_K \\
\text{NP}_L \\
\text{NP}_M \\
\text{NP}_N \\
\text{NP}_O \\
\text{NP}_P \\
\text{NP}_Q \\
\text{NP}_R \\
\text{NP}_S \\
\text{NP}_T \\
\text{NP}_U \\
\text{NP}_V \\
\text{NP}_W \\
\text{NP}_X \\
\text{NP}_Y \\
\text{NP}_Z \\
\text{NP}_a \\
\text{NP}_b \\
\text{NP}_c \\
\text{NP}_d \\
\text{NP}_e \\
\text{NP}_f \\
\text{NP}_g \\
\text{NP}_h \\
\text{NP}_i \\
\text{NP}_j \\
\text{NP}_k \\
\text{NP}_l \\
\text{NP}_m \\
\text{NP}_n \\
\text{NP}_o \\
\text{NP}_p \\
\text{NP}_q \\
\text{NP}_r \\
\text{NP}_s \\
\text{NP}_t \\
\text{NP}_u \\
\text{NP}_v \\
\text{NP}_w \\
\text{NP}_x \\
\text{NP}_y \\
\text{NP}_z \\
\text{NP}_A \\
\text{NP}_B \\
\text{NP}_C \\
\text{NP}_D \\
\text{NP}_E \\
\text{NP}_F \\
\text{NP}_G \\
\text{NP}_H \\
\text{NP}_I \\
\text{NP}_J \\
\text{NP}_K \\
\text{NP}_L \\
\text{NP}_M \\
\text{NP}_N \\
\text{NP}_O \\
\text{NP}_P \\
\text{NP}_Q \\
\text{NP}_R \\
\text{NP}_S \\
\text{NP}_T \\
\text{NP}_U \\
\text{NP}_V \\
\text{NP}_W \\
\text{NP}_X \\
\text{NP}_Y \\
\text{NP}_Z \\
\text{NP}_a \\
\text{NP}_b \\
\text{NP}_c \\
\text{NP}_d \\
\text{NP}_e \\
\text{NP}_f \\
\text{NP}_g \\
\text{NP}_h \\
\text{NP}_i \\
\text{NP}_j \\
\text{NP}_k \\
\text{NP}_l \\
\text{NP}_m \\
\text{NP}_n \\
\text{NP}_o \\
\text{NP}_p \\
\text{NP}_q \\
\text{NP}_r \\
\text{NP}_s \\
\text{NP}_t \\
\text{NP}_u \\
\text{NP}_v \\
\text{NP}_w \\
\text{NP}_x \\
\text{NP}_y \\
\text{NP}_z \\
\text{NP}_A \\
\text{NP}_B \\
\text{NP}_C \\
\text{NP}_D \\
\text{NP}_E \\
\text{NP}_F \\
\text{NP}_G \\
\text{NP}_H \\
\text{NP}_I \\
\text{NP}_J \\
\text{NP}_K \\
\text{NP}_L \\
\text{NP}_M \\
\text{NP}_N \\
\text{NP}_O \\
\text{NP}_P \\
\text{NP}_Q \\
\text{NP}_R \\
\text{NP}_S \\
\text{NP}_T \\
\text{NP}_U \\
\text{NP}_V \\
\text{NP}_W \\
\text{NP}_X \\
\text{NP}_Y \\
\text{NP}_Z \\
\text{NP}_a \\
\text{NP}_b \\
\text{NP}_c \\
\text{NP}_d \\
\text{NP}_e \\
\text{NP}_f \\
\text{NP}_g \\
\text{NP}_h \\
\text{NP}_i \\
\text{NP}_j \\
\text{NP}_k \\
\text{NP}_l \\
\text{NP}_m \\
\text{NP}_n \\
\text{NP}_o \\
\text{NP}_p \\
\text{NP}_q \\
\text{NP}_r \\
\text{NP}_s \\
\text{NP}_t \\
\text{NP}_u \\
\text{NP}_v \\
\text{NP}_w \\
\text{NP}_x \\
\text{NP}_y \\
\text{NP}_z \\
\text{NP}_A \\
\text{NP}_B \\
\text{NP}_C \\
\text{NP}_D \\
\text{NP}_E \\
\text{NP}_F \\
\text{NP}_G \\
\text{NP}_H \\
\text{NP}_I \\
\text{NP}_J \\
\text{NP}_K \\
\text{NP}_L \\
\text{NP}_M \\
\text{NP}_N \\
\text{NP}_O \\
\text{NP}_P \\
\text{NP}_Q \\
\text{NP}_R.
A Multi-Modal Combinatory Categorial Grammar analysis of Japanese Nonconstituent Clefting

(8) a. Ken ga Mari ni sono hon o yon-de morat-ta.
Ken NOM Mari DAT that book ACC read-MKR BENEF-PAST
‘Ken had Mari read that book for him.’

b. [Ken ga yon-de morat-ta] no wa Mari ni
Ken NOM read-MKR BENEF-PAST NMLZ TOP Mari DAT
sono hon o da.
that book ACC COP
lit. ‘What Ken had read for him was Mari that book.’

c. *[Ken ga Mari ni morat-ta] no wa sono hon
Ken NOM Mari DAT BENEF-PAST NMLZ TOP that book
o yon-de da.
ACC read-MKR COP
intended: ‘What Ken had Mari do for him was read that book.’

Here, (8a) is an example of a non-clefted sentence. (8b) is a case in which the embedded accusative object and the matrix dative object are clustered together in the focal position; this example shows that the pattern of clefting in this complex predicate construction is fairly flexible. Basically, as long as the embedded verb (V1) and embedding verb (V2) are not separated from one another, the sentence is grammatical. Example (8c) reveals (perhaps somewhat surprisingly) that it is indeed ungrammatical to cleft the entire embedded verb phrase, but just as in (8d), the ungrammaticality is due to the separation of V1 and V2. In order to account for these data, we crucially distinguish two modes: the normal (or ‘scrambling’) mode, which is permutative and associative, and the complex predicate mode, which is neither permutative nor right associative, but rather is only left associative.\(^6\)

We describe the foundations of such a system below before turning to a demonstration of how this analysis captures the facts above.

4.1 The Formal System

The following is the hierarchy of the modes we will employ:\(^7\)

(9)

---

\(^6\)The existence of these modes is independently motivated to account for further properties of complex predicates, such as scrambling of arguments of the embedded predicate with those of the higher predicate. Unfortunately, space limitations preclude us from discussing these properties here.

\(^7\)Here and elsewhere in this section, we follow the general approach of Baldridge (2002) but differ somewhat in detail in order to assume the minimum theoretical machinery necessary for accounting for the Japanese data. We have no reason not to think that the analysis presented here could be reformulated in Baldridge’s system.
The modes are arranged from top to bottom by their permissibility; the top node (∗) is the least permissive and is neither permutative nor associative in either direction, while the bottom node (·) is the most permissive and is both permutative and associative in both directions. The three modes bearing intermediate permissibility each have a single property: (∝) is left associative, (∝) is right associative, and (×) is permutative.

The distinction between right and left associative modes (which is not present in Baldridge’s system) is introduced here in order to distinguish two ‘restructuring’ operations corresponding to the following binary structural rules in Type Logical Grammar (TLG) (Oehrle, 1998):

\[
\begin{align*}
(10) & \quad \text{a. Right Association} \\
& \quad A \cdot_{\infty} (B \cdot_{\infty} C) \quad (A \cdot_{\infty} B) \cdot_{\infty} C \\
& \quad (A \cdot_{\infty} B) \cdot_{\infty} C \quad A \cdot_{\infty} (B \cdot_{\infty} C)
\end{align*}
\]

These TLG rules will be incorporated into the MMCCG system by revising the FC schema from the last section in the following way:

\[
\begin{align*}
(11) & \quad \text{a. } A /_{\infty} B \ B /_{\infty} C \vdash A /_{\infty} C \quad \text{b. } A \ \backslash_{\infty} B \ C \ \backslash_{\infty} A \vdash C \ \backslash_{\infty} B
\end{align*}
\]

The modality specification here ensures that two functors can be composed only when the modality of each (in addition to the directionality of the slashes) matches. The distinction of left and right associative modes is motivated by empirical evidence: as we will see below, by assigning the left associative mode as the combinatoric mode for the complex predicate formation, the syntactic properties of the -te form complex predicate can be neatly captured, including the pattern it exhibits when it interacts with clefting. As for the other combinatoric rules from (5), FA remains unchanged except that it is specified for the least permissive ∗ mode (which ensures that it is applicable to any mode as guaranteed by the convention of rule schema application described in footnote 9). TR requires the following slight revision in order to guarantee that the original combinatorial property is preserved after type-raising.\(^{10,11}\)

---

\(^8\)We have limited ourselves to modelling the effects of binary structural rules of association and permutation in Oehrle (1998) in CCG, whose linguistic motivations are better understood than unary structural rules. For more recent and detailed discussion of formal characterizations and empirical applications of structural rules in TLG, see Moortgat (1996), Bernardi (2002) and Vermaat (2005).

\(^9\)More precisely, following Baldridge, we assume that combinatoric rules can apply only when the modality specification on the input is more permissive than what is specified in the rule. For example, (11a) is applicable when the modality of the slash of the lefthand element of the input (i.e. what instantiates A /_{\infty} B) is the most permissive mode (·).

\(^{10}\)We employ the following convention: any slash without a specified modality is an abbreviation of / or \, the most permissive mode.

\(^{11}\)The index i is a variable notation for slash modalities. The purpose of this variable index here is to ensure that the modality specifications match for both slashes when a category is type-raised over another category.
A Multi-Modal Combinatory Categorial Grammar analysis of Japanese Nonconstituent Clefting

(12) a. \( A \vdash B/i(B\i_A) \) b. \( A \vdash B\i(B/i_A) \)

In addition to these rules, we introduce one nonlogical unary rule to handle scrambling:\(^{12,13}\)

(13) a. \( A/x B/x C \vdash A/x C/x B \) b. \( A\i_x B\i_x C \vdash A\i_x C\i_x B \)

This enables a functor looking for two categories successively in the same direction (and in the permutative mode) to flip the order of these arguments. This allows each verb to be listed only once in the lexicon with its basic word order, since all other orders can be obtained from that basic entry by successive applications of (13). The following, then, are the lexical entries necessary for the derivations of the relevant examples.\(^{14,15,16}\)

(14) a. morat-ta: \( S\i_{NP} \i_{NPd}\_\infty VP \) b. yon-de: \( VP\i_{NPd} \)

As we have alluded to all along, the benefactive verb morat-ta takes its embedded verb argument in the left associative or ‘complex predicate’ mode. We also see that the prenominal adjective nagai ‘long’ takes its nominal argument in the least permissive mode, indicating the tightest possible relationship between two lexical items.

4.2 Accounting for the Data

We begin by demonstrating how the new system handles grammatical cases of nonconstituent clefting with complex predicates, such as (8b).

\(^{12}\) The semantics for these permutation rules can be defined as follows:

(1) a. \( A/x B/x C \vdash \lambda x_0…x_n y z. \varphi \vdash A/x C/x B \vdash \lambda x_0…x_n y z. \varphi \)

With these definitions, the straightforward syntax-semantics interface of CCG in K&S’s analysis is maintained. For the semantics of other rules and lexical entries and sample derivations illustrating the syntax-semantics interface assumed here, the reader is referred to Kubota and Smith (2006).

\(^{13}\) An important alternative to this approach is set-based CCG (Hoffman, 1995), as pointed out by an anonymous reviewer. While the analysis of word-order in this paper in terms of a unary permutation rule might be seen as introducing too much flexibility in the grammar, we have opted for this account for the following reasons: (i) it allows one to capture the relevant linguistic generalizations relatively straightforwardly and (ii) it maintains the straightforward syntax-semantics interface of standard CCG in which model-theoretic interpretation is directly obtained from surface composition, which is lost in the multi-set CCG alternative. A detailed comparison of the present proposal and the set-based CCG alternative is beyond the scope of this paper.

\(^{14}\) VP is an abbreviation of \( S\i_{NPd} \).

\(^{15}\) Entries for no, wa and da are unchanged; slashes will be specified in the \( \cdot \) mode.

\(^{16}\) The verb morat-ta is ambiguous between its use as a benefactive predicate and ordinary lexical verb meaning ‘receive’. We assume that there is a different ditransitive verb entry for the latter.
Here, the V1 and V2 can function compose since the slash modality specifications on the rule subsume those lexically specified for the verbs. From this point on, the cluster of the V1 and V2 effectively functions as a single ‘ditransitive verb’ looking for arguments of the embedded verb and the higher verb successively. Permutation is then applied to scramble the order of the verbs’ arguments in order to combine next with the subject Ken ga, and the rest of the derivation proceeds in parallel to the example from K&S.

The next two derivations demonstrate the ability of our system to block the ungrammatical examples seen in (8c) and (8d). In (8c), the dative matrix argument appears linearly adjacent to the matrix verb. Thus, in order for these words to combine, the order of the dative NP and the embedded VP in the lexical specification of the matrix verb needs to be flipped so that the dative NP becomes the first argument that the matrix verb is looking for. However, the permutation rule (13) cannot be applied here due to the tight connection between the embedded and embedding verbs encoded in the lexical entry of morat-ta with the left associative mode (⊳) as illustrated in the following failed derivation:

\[(16)\]

In (3a), on the other hand, it is not permutation that causes the problem. In this case, the fact that the \(\ proprio\) slash modality for the embedded verb is left associative but not right associative plays a crucial role. In order for (3a) to be derived, the clefted string would have to be analyzed as a constituent that is looking for the matrix verb to become an S. This means that, in the focal position, the embedded VP would have to be type-raised over \(S\backslash N\backslash P\) so that it could successively function compose with the matrix nominative and dative arguments to result in the desired category. But in order to match the lexical specification on the matrix verb (so that it can eventually combine with it), it would have to type-raise with the \(\ proprio\) modality. Thus, the embedded VP is type-raised to the category \((S\backslash N\backslash P)\backslash(\proprio)(S\backslash N\backslash P)\backslash(\proprio)\), as we see in (17):

\[17\]
A Multi-Modal Combinatory Categorial Grammar analysis of Japanese Nonconstituent Clefting

(17) Mari ni $\frac{NP}{S}$ $\frac{d}{NP}$ TR

$\frac{(S\backslash NP_n) / ((S\backslash NP_n) \backslash NP_2)}{S\backslash NP_n}$ TR

$\frac{sono\ hon\ o\ yon-de}{VP}$

$\frac{((S\backslash NP_n) \backslash NP_2) / Perm ((S\backslash NP_n) \backslash NP_2) \backslash Perm \ VP}{(S\backslash NP_n) \backslash Perm \ VP}$ TR *FC

Being in this new category, however, the embedded VP cannot function compose by (11a) with the type-raised matrix dative argument since $\bowtie$ isn’t right associative, and so the derivation fails, ruling out the sentence which K&S’s analyses overgenerates.

Finally, in prenominal modification, the ungrammatical example in (4b) that was overgenerated by K&S’s system is also blocked in our analysis:

(18) hon $\frac{a}{NP_a}$ $\frac{o}{NP_a}$ TR

$\frac{Taro\ ga\ yon-da}{S\backslash NP_n}$

$\frac{S\backslash NP_n}{Perm}$

$\frac{NP_n}{NP_n}$

$\frac{Perm}{Perm}$ $\langle$ Perm $\rangle$

In this example, in order to derive the topicalized sentence, the noun hon $o$ ‘book’ and the sentence Taro $ga$ yon-da ‘Taro read’ need to combine via FC. However, due to the mismatch of slash modality, the derivation is blocked (crucially, hon $o$ has to first type-raise with the $*$ modality in this example so that it would match the lexical specification of the adjective nagai ‘long’ and ultimately combine with it). Thus, we have seen that the addition of a minimal number of modes enables the structural control necessary to correctly derive all of the grammatical examples while blocking all of the ungrammatical examples.

5 Conclusion

Though K&S’s CCG analysis of Japanese nonconstituent clefting succinctly captures the essential flexibility of clefting in Japanese, we have seen that further restrictions are necessary to accurately model the full range of data. Adopting MMCCG and positing a few slash modalities independently-motivated to treat complex predicates, the analysis proposed here accurately accounts for all of the data. In addition, these modalities enable the introduction of the permutative rule, which accounts for Japanese scrambling phenomena in a more elegant way than in the previous approach. Given that this analysis effectively describes the interaction between clefting, scrambling, and complex predicates in Japanese, we hope that it will serve as a springboard for further investigations into a more complete and integrated account of Japanese syntax.\footnote{As an anonymous reviewer points out, an interesting question for future research is to see whether an account of a wider range of scrambling phenomena is possible in the present setup of MMCCG; it is known that clause-internal scrambling and long-distance scrambling behave differently in terms of phenomena such as quantifier scope. For an overview of scrambling phenomena in Japanese and relevant literature, see Nemoto (1999).}
Bibliography


Extracting Predicates Subcategorizing for Wh-Clauses: an Architecture for a Semi-automatic System

EKATERINA LAPSHINOVA
Institut für maschinelle Sprachverarbeitung (IMS)
University of Stuttgart
ekaterina.lapshinova@ims.uni-stuttgart.de

ABSTRACT. This paper describes an architecture for automatic extraction and semi-automatic classification of predicates subcategorizing wh-clauses in German, which was developed as a part of broader research on automatically extracted lexical data. The main goal is to develop a semi-automatic classification system for lexical data, in particular German verbs, taking into account their subcategorisation frames, as well as their morpho-syntactic features, e.g. tenses or collocational preferences. The paper gives a preliminary overview of the developed architecture and describes first results, problems to be solved and tasks ahead.

1 Introduction

The number of NLP technologies for lexicon acquisition which are aimed at high quality results has been growing in the recent years. The reason for this is that creating lexical information resources manually costs much time and effort. The lexical information retrieved with acquisition tools can be stored in machine-readable lexicons and updated dynamically [Schulte im Walde 2006]. Many such tools aim at extracting words with their linguistic properties and tend to classify them syntactically or semantically.

This study presents a preliminary set of automatic extraction procedures, which are part of a broader research programme about the semi-automatic classification of automatically extracted language data, in particular verbs and their subcategorisation. The research aims at contributing to the creation of subcategorisation lexicons for German and to the enhancement of existing ones, for example IMSLex [Lezius/Dipper/Fitschen 2000]. As the lexical data are created to serve symbolic grammars (e.g. a German LFG grammar, [Forst 2003]), we have opted for non-probabilistic extraction methods.
Extracting Predicates for Wh-Clauses

In this paper, we deal with sentential complements focusing on predicates which subcategorize a wh-clause although our methods can be applied to other complements as well. The context for the extraction of such constructions are German verb-final sentences (i.e. mainly subclauses, ca. 20-25% of all corpus text), as we expect less noise in sentences of this constituent order type than in the two others: 62.6% of random sample of 500 verb-final sentences with wh-complements have a subcategorized wh-clause, whereas this ratio is of only 43.2% in verb-second sample of the same size. In addition, the Mittelfeld (middle-field) in verb-second clauses contain more diverse valency bearer (57% verbs, 33.3% nouns, 8.3% adjectives), whereas predicates in verb-final clauses are more homogeneously verbal (85% vs. 7.3% nouns and 6.4% adjectives) and thus preferred for verb subcategorization extraction. Finally, the sequence of elements in verb-final clauses is more regular: the subcategorized subclause typically follows the verb, and nominal or adjectival valency carriers tend to proceed it and can be easily “detected” on the left side of the verb in a verb-final sentence.

The extraction is planned to capture not only subcategorisation patterns, but also contextual preferences\(^1\) of the respective verbs, e.g. for tenses or specific collocations.

In the following two sections we present the data we worked with and describe the modules of extraction architecture we used illustrating it with examples. In the last sections we evaluate the results, conclude and suggest further development.

2 Data and Tools

2.1 Corpus

We used a corpus of newspaper texts of Frankfurter Rundschau (‘FR’, 40M tokens), Frankfurter Allgemeine Zeitung (‘FAZ’, 70M tokens) and Tageszeitung (‘taz’, 111M tokens) which are a part of a collection of German newspaper corpora (ca. 300M tokens) available at the IMS. The corpus is annotated with part-of-speech tags, morpho-syntactic information, lemmatisation and partial syntactic analysis results\(^2\), and can be queried with CQP, Corpus Query Processor [Evert 2005]. Queries may be regular expressions over arbitrary configurations of word form strings and annotations. For example, query 1 in Figure 1.1 is aimed at a sequence of a finite form of the lemma \texttt{fragen} (“to ask”), a comma and a relative pronoun. It approximates parts of verb final sentences with indirect interrogatives after a finite verb).

\(^1\)Contextual preferences, especially morpho-syntactic preferences of noun-verb-collocations are discussed in [Heid 2005], [Ritz/Heid 2006] and [Evert/Heid/Spranger 2004]\

\(^2\)Annotations were obtained with Tree Tagger [Schmid 1994] and [Schmid 1999], YAC-Chunker [Kermes 2003] and the morphology tool SMOR [Schmid/Fitschen/Heid 2004]
Ekaterina Lapshinova

query 1: [lemma "fragen"&pos="V.FIN";"[pos="PW.*"]];
search results for query 1:
1896347: Herstellers . Die Polizei <fragt , wer> ein solches Fahrrad gese
2659907: Wenn sich jetzt jemand <fragt , warum> er sich überhaupt
8924905: nalisten sorgten sich und <fragen , welcher> Drogenhändler.

Figure 1.1: A sample of CQP-query for the verb *fragen* ("to ask")

2.2 Types of extracted predicates

For the verb valency extraction task, we applied sequences of complex CQP-
queries which are the building blocks of the developed retrieval architecture.
Different subcorpora are extracted from the newspaper corpus according
to three types of predicates whose subcategorisation is to be explored: a
subcorpus of predicative adjectives, a subcorpus of noun+verb- multiword
expressions and a subcorpus of verbs.

This choice is based on the assumption that the following parts of speech
are able to have subcategorized complements in a sentence: verbs, nouns and
adjectives. Although some linguists who investigate the notion of valency
mention the predicative nature of all three parts of speech, most of them con-
centrate exclusively on verbal predicates. Referring to existing dictionaries
of valency
\footnote{Works of [Herbst et al. 2004], [Sommerfeldt/Schreiber 1983a], [Sommerfeldt/Schreiber 1983b],
[Sommerfeldt/Schreiber 1996].}
we focus on the three classes of predicates mentioned above.

The subcorpus of predicative adjectives contains constructions consisting
of a predicative adjective combined with a verb or a verbal complex, as
illustrated in (1). The dependent wh-clause (underlined in the example),
which is extracted along with the valency carrier (printed in bold in the
example), is part of the valency of the adjective concerned.

\begin{align}
(1) & \text{[...]} \text{daß noch immer nicht absehbar ist, wann der Treffpunkt[...].}
& \text{("[...][that it is still unforeseeable, when the meeting place[...]").} \\
& \text{Multiword expressions (mwe) are classified into two subsets of noun+verb}
& \text{combinations\footnote{The definition of these two multiword expression types is based on Perssons classifi-
\text{cation [Persson 1975].}}: mwe1, in which a wh-clause is subcategorized by a nominal}
& \text{predicate (which appears in a support verb constructions - example (2)) or}
& \text{mwe2 - where the whole expression subcategorizes for a wh-clause (3).}

\begin{align}
(2) & \text{Wenn also die Frage gestellt wird, wo Erziehung stattfindet[...].}
& \text{("If one poses the question, where education takes place[...]").} \\
(3) & \text{Wenn die nun aber ins Grübeln kommen, wen sie[...].}
& \text{("If they just ponder (come into pondering), whom they[...]").} 
\end{align}
Extracting Predicates for Wh-Clauses

The subcorpus of verbs consists of phrases containing valency carrier verbs, where extracted subordinate clauses are a part of the verb valency, as shown in (4).

As far as verbal predicates are concerned, we intend to detect not only their wh-sentential complements, but also other parts of their subcategorisation frames and their morpho-syntactic features. We distinguish reflexive vs. non-reflexive verbs, and, as an orthogonal dimension, verbs with vs. without a “Korrelat” (a pronominal adverb, e.g. *darüber* in (9a.), below).

(4) Sie zeigten mir, da saßen Leute, die *wußten*, worauf es ankommt. (*They showed me: there were people, who knew, what it depends on*).

The described types of predicates led us to distinguish two levels of extraction:

1st level (predicate types):
1. predicative verbs + adjectives;
2. mwe: verbs + nouns;
4. verbs.

2nd level (additional properties):
a. non-reflexive without a Korrelat;
b. reflexive without a Korrelat;
c. non-reflexive with a Korrelat;
d. reflexive with a Korrelat.

The second-level-extraction procedure is applied to both the extracted subcorpus of predicative adjectives and the subcorpus of verbs obtained in the first-level-extraction. The extraction of mwes of both classes (cf. (2) and (3)) occurs at the first level. The reason for it is our method of cascaded extraction: from general queries for verb-final constructions (Figure 1.2) to more specific ones (Figure 1.3). This saves much effort and time in further evaluation.

3 Architecture

3.1 Extraction and filtering of predicates with wh-clauses

Our extraction architecture consists of three components: procedures for the general extraction of predicates subcategorizing wh-clauses, and the implementation of the two levels of extraction defined in section 2.2. The extraction steps proceed from the general to the specific.

In the first step, we extract complex sentences consisting at least of a main clause and one subordinate wh-clause (Figure 1.2). The predicate whose subcategorisation we are exploring (line 3 in Figure 1.2) is followed by a comma (line 4) and a wh-word (line 5) which introduces a subordinate clause (lines 5-8). The basic assumption is that the extracted subordinate wh-clause is subcategorized by the verb or by other elements (adjectives or nouns) which are mostly immediately followed by the verb. An example of
Figure 1.2: Query for predicates in a verb-final phrase subcategorizing a wh-clause as in (4)

an extracted sentence is given in Figure 1.2. The extracted cases are saved as a subcorpus (Extraction 1 and Subcorpus 1 in Figure 1.4).

Then, we remove “noise”: phrases which are mostly cases of headless relatives or adverbial clauses (Filtering in Figure 1.4). For example we cut out sentences which contain such words like dort, da, dahin, etc. or das, alles, etwas etc. in the clause of the valency carrier and wo, wohin or was in the subordinate clause, as illustrated in (5).

(5) Weil ohnehin nur das richtig ist, was Siege [...] bringt. (“Because only that is appropriate, what provides victories[...]”).

3.2 Implementation of two levels of extraction: subcorpora and subsets

Later the filtered set of candidate sentences (Subcorpus 2 (SC2) in Figure 1.4) is partitioned into several subcorpora according to the predicate types described in section 2.2. The subdivision into further subsets is carried out in the extraction process (Figure 1.4 Extraction 2).

1st level. We extract a subcorpus of predicative adjectives first (Subcorpus 3 (SC3) in Figure 1.4). The query to extract adjectival predicates is built up of similar elements as the general one (Figure 1.2). We just replace line 2 in Figure 1.2 with a building block for a predicative adjective (line 1 in Figure 1.3). The search is applied to Subcorpus 2 in Figure 1.4 and, in this case we search for a predicative adjective among the words matching line 2 in Figure 1.2. The cases found here are saved as a subcorpus to be used for the second level extraction.

For the search for mwel1 and mwel2 described in 2.2 (Subcorpus 4 (SC4)
and Subcorpus 5 (SC5) in Figure 1.4), we define two different queries for the left part of the sentence to be extracted. The first query contains a noun that subcategorizes a wh-clause, and a verb (2). The nouns\(^5\) are known from IMSLex [Lezius/Dipper/Fitschen 2000]. This kind of nouns is sometimes found in the Vorfeld-position (pre-field position) together with its sentential complement, as in (6). The occurrence of this predicate type in corpora can be tested by a query, which contains a building block for the noun at the sentence beginning followed by a wh-clause and a finite verb.

\[(6)\] Die logische Frage, wo die Autofahrer nunmehr sind, stellte sich[...] ("The logical question, where the drivers are now, [...]asked itself").

The query for the second type of multiword expressions, illustrated in (3), is built up of similar elements. It starts with a preposition or a combination of a preposition with an article (e.g.\(\text{in+das=}\text{ins}\)), followed by a noun and a given verb from a list of frequent support verbs\(^6\). The differences of both queries are presented in Table 1.1.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
query & building blocks & comments & matching sentence \\
\hline
1. & \(\text{[pos="ADJD"]}\) & predicative adjective & wenn ich sehr \\
2. & \(<vc>\ldots</vc>\) & verbal complex & interessant \\
3. & \(\ldots\) & comma & finde \\
4. & \([\text{pos="PW.*"}]\) & wh-clause-start: relative pronoun & was \\
5. & \([\text{pos="V.*FIN"}]\) & optional words, no finite verbs & \(\ldots\) \\
6. & \([\text{pos="V.FIN"}]\) & finite verb & machen \\
7. & \([\text{pos="$\.$}]\) & sentence end & . \\
8. & within s; & within a sentence & sentence context \\
\hline
\end{tabular}
\caption{Query for predicative adjectives subcategorizing a wh-clause.}
\end{table}

Table 1.1: Differences in the query building blocks for mwe1 and mwe2.

The last step of the first-level extraction procedure is to define the verbal subcorpus and its subsets. Subcorpus 6 (SC6) (Figure 1.4) containing all kinds of verbal predicates is obtained by subtracting the extracted subcorpora described above from SC2 (Figure 1.4):

\(^5\)Nouns: e.g. Antwort, Frage, Diskussion, Entscheidung, Information, etc.
\(^6\)Verbs: bringen, fallen, geben, gehen, gelangen, geraten, kommen, machen, setzen, treten, ziehen. The verb list is based on studies on “Funktionsverbgefüge” (support verb constructions) in [Breidt 1993], [Langer 2005], [Persson 1975].
SC 6 = SC 2 − (SC 3 + SC 4 + SC 5).

2nd level. On the second level of extraction we specify queries for sub-
sets within Subcorpus 3 and Subcorpus 6 (Figure 1.4) which are aimed at
searching for cases with a reflexive pronoun and for those with a “Korrelat”. For
this purpose, we include a query building block with either a reflexive
pronoun or a “Korrelat” in front of the adjective or the verb concerned. These
cases, illustrated in (7) and (8), are extracted from of Subcorpus 3 and 6 and saved as +reflexive/-“Korrelat” and -reflexive/+“Korrelat”, cf.
Figure 1.4. Then we search for cases with both a “Korrelat” and a re-
flexive pronoun (cf.(9)) within the subsets with a “Korrelat”, by querying reflexive pronouns in front of the verb or the predicative adjective (subset:
+reflexive/+“Korrelat” in Figure 1.4). The -reflexive/-“Korrelat” cases are
obtained like Subcorpus 6 by subtracting the extracted subsets from Sub-
corpus 3 or Subcorpus 6.

Figure 1.4: Extraction architecture.

(7) a. [.../dass [...]Kunstsammler sich bewusst bleiben, woher [...].
(“[...]that [...]art collectors stay (themselves) conscious
(about), where [...].”)

b. [...]ohne sich vorher zu vergewissern, welches Stück [...]läuft”.
(“[...]not ascertaining (themselves) what play is being given”).

(8) a. [...]dass die Stadt [...]nicht alleine dafür zuständig ist, wo [...].
(“[...]that [...]is not the city alone responsible for this, where [...].”)
Extracting Predicates for Wh-Clauses

b. Ob ein Vertragsabschluß davon abhängt, wann [...] (“Whether the conclusion of the agreement depends on, when [...]”).

(9) a. [...] dass viele Menschen [...] sich auch darüber unsicher sind, wer oder was sie sein wollen. (“[...] that many people [...] are (themselves) not sure (about), who or what they want to be”).

b. Wer sich mit uns darüber unterhalten will, was [...]. (“Who wants to converse (themselves) with us about, what [...]”).

4 Results

The extraction architecture described above can be applied to any type of German corpora annotated as stated above in section 2.1. We tried it on ‘FR’, ‘FAZ’ and ‘taz’. The extraction results from other corpora served as a test for cases where the matches brought few results. The number of extracted phrases for each subcorpus is shown in Table 1.2. The frequency of some extracted samples of predicates are presented in Table 1.3.

<table>
<thead>
<tr>
<th>extracted from</th>
<th>found</th>
<th>pred.adjs</th>
<th>mwe1</th>
<th>mwe2</th>
<th>verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘FR’</td>
<td>1880</td>
<td>82</td>
<td>61</td>
<td>23</td>
<td>1763</td>
</tr>
<tr>
<td>‘FAZ’</td>
<td>3436</td>
<td>184</td>
<td>188</td>
<td>40</td>
<td>3034</td>
</tr>
<tr>
<td>‘taz’</td>
<td>7214</td>
<td>314</td>
<td>267</td>
<td>73</td>
<td>6585</td>
</tr>
</tbody>
</table>

Table 1.2: Extraction results (frequency).

<table>
<thead>
<tr>
<th>preds</th>
<th>pred.adjs</th>
<th>verb: fragen</th>
<th>mwe1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>klar</td>
<td>unklar</td>
<td>deutlich</td>
</tr>
<tr>
<td>‘FR’</td>
<td>12</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>‘FAZ’</td>
<td>25</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>‘taz’</td>
<td>41</td>
<td>25</td>
<td>13</td>
</tr>
</tbody>
</table>

Table 1.3: Selected extraction results.

Our extraction architecture delivers promising results, but the extracted data still contain “noise”- and “silence”-cases. Most of the “noise”-cases are phrases with headless relatives as well as adverbial relative clauses. Their form doesn’t differ from that of the cases relevant for this study ((10a.) vs. (10b.)), and therefore their detection in extracted results is problematic. Some adverbial relatives “survive” the filtering procedures (cf. section 3.1).

7German newspaper corpora: ZEIT (‘ZEIT’, ca. 40M tokens) and Handelsblatt (‘HB’, ca.36M tokens)
Some queries deliver very few results, especially for the extraction of multiword expressions or cases with both a reflexive pronoun and a “Korrelat”, because of low frequency. But mostly these are cases of “silence”. The sentences in (11) should have been extracted as mwe1 (Figure 1.4) but were not matched by queries. Both the nouns and the verb are not contained in the existing lists: the words Überblick (“overview”) in (11a) or Klarheit (“clarity”) in (11b) were not found in the IMSLex noun list (see 3.2) and the word verschaffen (“to provide/make/get”) is not included in the list of support verbs. Experiments to enhance the list of nominal predicates are under way [Heid/Lapshinova 2007].

Furthermore, some cases remained “undetected” due to the incomplete definition of the applied queries. For instance, the query for the subset of reflexive predicates should be enhanced to make it possible to detect not only sich-cases (a 3rd person form) but also 1st and 2nd person form cases (cf. (12)).

Sometimes the predicates subcategorizing wh-clauses are misclassified. The phrase in (13) extracted as a verbal predicate with a reflexive pronoun and a “Korrelat” is in fact an idiom, in which the wh-clause is subcategorized by the whole phrase, and not by the verbal predicate.

First quantitative evaluations of the data from Tables 1.2 and 1.3 presented in tables below (Tables 1.4 and 1.5) show that our method is good for extracting all kinds of predicates (verbs: precision of ca. 60%, predicative adjectives: precision of ca. 60-70%, recall of ca. 95% or nominal predicates - mwe1: precision of ca. 60-70% and recall of 56%), except mwe2. Accuracy for this predicate type is much lower (recall of under 10-20%).
Extracting Predicates for Wh-Clauses

<table>
<thead>
<tr>
<th>preds</th>
<th>pred.adjs</th>
<th>mwe1</th>
<th>mwe2</th>
<th>verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>corpora</td>
<td>FR</td>
<td>FAZ</td>
<td>taz</td>
<td>FR</td>
</tr>
<tr>
<td>prec.(%)</td>
<td>68,2</td>
<td>64,6</td>
<td>73,5</td>
<td>67,2</td>
</tr>
</tbody>
</table>

Table 1.4: Precision assessed on some types of predicates.

<table>
<thead>
<tr>
<th>preds</th>
<th>pred.adjs</th>
<th>mwe1</th>
<th>mwe2</th>
</tr>
</thead>
<tbody>
<tr>
<td>recall(%)</td>
<td>94,9</td>
<td>56,2</td>
<td>50</td>
</tr>
<tr>
<td>prec.(%)</td>
<td>68,2</td>
<td>67,2</td>
<td>17,3</td>
</tr>
<tr>
<td>f-score(%)</td>
<td>79,3</td>
<td>61,2</td>
<td>25,7</td>
</tr>
</tbody>
</table>

Table 1.5: Recall, precision and f-score assessed on predicates from ‘FR’.

5 Conclusion and future work

We elaborated an architecture for extracting and classifying different types of predicates subcategorizing wh-clauses in German. The extraction is a process of an increasingly refined subclassification of the corpus data. It is built up of stepwise procedures from predicates with wh-clauses over a subdivision according to the kinds of predicates at hand, to subsets according to morpho-syntactic properties of the subcategorisation frames of the predicates. The first extraction results are promising, even though an increase in precision is still required.

Current tests with German newspaper corpora (cf. section 4) show the need for further refinement of the queries, e.g. by inserting building blocks for case and other morpho-syntactic properties. This will allow us to capture the full subcategorisation frames (for instance, <Subject, Indirect-Object, wh-clause> in (11b.), where the current query misses out on the indirect object), as well as tense and mood of the verbal predicates analysed. Moreover, we apply the same extraction model for the acquisition of other subordinate clause types, (that- and if-clauses): the first tests deliver less “noise” than the extraction of wh-clauses. For instance, “noise” from headless relatives and adverbial clauses (cf. section 4) only occurs with wh-clauses.

As mentioned above (cf. section 1) future work will capture other types of complements and include the extration of contextual preferences.

Bibliography


Ekaterina Lapshinova


Extracting Predicates for Wh-Clauses


Acquisition of Irregular Patterns in Spanish Verbal Morphology

LAIA MAYOL
University of Pennsylvania∗
laia@ling.upenn.edu

ABSTRACT. This paper examines acquisition data of two irregular patterns in Spanish verbal morphology: diphthongization and velar insertion. The data contains 345 instances of errors, which are not equally distributed among all conjugation classes or error types. A statistical analysis revealed that input from adults and whether a verb belongs to the first conjugation are the variables that best predict the verbs’ correct usage. This data poses challenges to theories of morphology that claim that all irregular verbs are individually stored in the lexicon, while it is compatible with theories that propose that rules apply to subclasses of irregular verbs.

1 Introduction

This paper examines the acquisition of some irregular patterns in Spanish verbal morphology, namely those involving diphthongization and velar insertion. I present data on the acquisition of these patterns, perform a statistical analysis and relate it to different models proposed to account for child acquisition of morphology, namely, the Words and Rules model (Pinker and Ullman 2002) and the Rules and Competition model (Yang 2002).

English past tense has been the object of a long controversy in the literature between connectionism and generative linguistics (see McClelland and Patterson (2002) and Pinker and Ullman (2002)). However, English past tense has a relatively simple morphology and phonology. Thus, the debate could benefit from looking at a morphologically more complex language, such as Spanish, in which there are clear patterns within the irregular verbs.

The paper is structured as follows: Section 2 describes the irregularities this paper is concerned with; Section 3 gives an overview of previous work on morphology acquisition; Section 4 presents the child data and the main results and Section 5 the analysis. Section 6 concludes.

∗Many thanks to Charles Yang for guiding me in this project and making the automatic data extraction possible. Thanks also to the three anonymous reviewers for their very detailed and useful comments and to Josh Tauberer for proof-reading this paper.
2 Irregular patterns in Spanish verbal morphology

Spanish verbs are highly inflected: their morphology combines inflectional pieces (which contain information about tense, mood, number and person agreement) with the stem; the stem can be further subdivided into a root and a theme vowel, as shown in (1) for the second person plural of the past tense of the verb *hablar* (‘to talk’):

(1) habla ba ais
    speak theme_vowel past 2p plural

There are three theme vowels in Spanish, which divide the verbs into three classes or conjugations: the first conjugation, marked with the vowel [a], is the open class conjugation and the one that contains most verb types; the second conjugation, marked with [e], and the third conjugation, marked with [i], have fewer members, an important number of which are irregular.

2.1 Diphthongization

Spanish verbs present a well-known morphophonemic alternation\(^1\). In certain verbs, mid vowels are diphthongized in stressed syllables. (2) shows the pattern for the present indicative of the verbs *comenzar* (‘begin’) and *contar* (‘count’).\(^2\)

(2) comienzo comienzas comienza comenzamos comenzáis comienzan
    cuento cuentas cuenta contamos contáis cuentan

This alternation is lexically arbitrary; that is, not all verbs containing mid vowels present a diphthong in stressed syllables. In fact, we find minimal pairs of verbs, such as *contar-montar*, in which the first verb presents the diphthong (*cuento*), while the second does not (*monto*).

In verbs from the first and second conjugation, this alternation is unpredictable. However, while in the first conjugation, diphthongization is a minority pattern and non-alternation is the default pattern for new verbs (Albright et al. 2000), in the second conjugation, diphthongization is more common. In contrast, all third conjugation verbs containing a mid vowel present an alternation. For such verbs, there are two different patterns: (i) diphthongization of the stressed syllables of the present indicative and (ii) raising of the stressed mid vowel (always to [i]). Both patterns are illustrated in (3), with the verbs *mentir* (‘lie’) and *pedir* (‘ask’), respectively.

(3) miento miéntes miénte mentimos mentís miéntan
    pido pídes pide pedimos pedís piden

---

\(^1\)See Harris (1985) for an analysis in which a phonological rule with a morphological conditioning derives the diphthongized forms.

\(^2\)The graphic accent in (2), (3) and (4) indicates phonological stress.
The diphthong alternation is not restricted to verbal morphology. De-
verbal nouns sometimes show the diphthong in the stressed syllable: juego
(‘game’ from jugar). This pattern is also found in pairs of related words, such
as bueno (‘good’) - bondad (‘goodness’) or huevo (‘egg’) - ovíparo (‘oviparous’).

2.2 Velar insertion

A velar stop is inserted in the first person singular of the indicative and also
in the subjunctive of some verbs from the second and third conjugation.
Although this alternation affects a small number of verbs, some of them are
very common, such as tener (‘to have’), poner (‘put’) or salir (‘go out’),
whose first person singular forms are tengo, pongo and salgo, respectively.

A few common verbs show both alternations (diphthongization and velar
insertion) at the same time. Both tener (‘to have’) and venir (‘to come’)
show velar insertion in the present subjunctive and 1st person singular of the
present indicative and [ie] diphthongization in 2nd singular and 3rd person of
the present indicative, as shown in (4) for the verb tener. Also, poner shows
velar insertion in the present subjunctive and first person singular of the
present indicative and [ue] diphthongization in the participle (i.e. puesto).

(4) tengo tién estén tienen tenemos tenéis tienen

3 Background

3.1 Morphology acquisition

Several models have been proposed to explain how children acquire regular
and irregular inflected forms. Here I am comparing two models which make
explicit use of rules:

1. The Words and Rules model (Pinker and Ullman 2002) proposes that
all regular forms are derived by a rule, while irregular forms are stored
in the lexicon. A stored form blocks the application of the rule (brought
blocks bringed). According to this approach, children produce overreg-
ularizations, such as goed, when the irregular form has not been stored
in the lexicon and, therefore, nothing prevents the rule from applying.
In contrast, overirregularizations are predicted to be impossible and
indeed, in English, they are extremely rare (Xu and Pinker 1995). This
approach denies that there are subclasses within the irregular verbs,
since every irregular verb is individually stored in memory.

Other models, such as connectionism, propose that there are no rules, but a network
which maps base forms to past-tense forms (Rumelhart and McClelland 1986). Since I
am mostly interested in models that make use of rules, I will not be discussing this model
further in this paper.

187
2. The Rules and Competition model (Yang 2002) assumes that there are rules both for the regular and for the irregular forms and that these rules compete against each other. Every irregular rule $R$ which applies to a certain verb class is associated with a probability $P_R$ and each assignment of a verb $x$ to an irregular class of verbs $S$ carries another probability $P(x \in S)$. The acquisition task consists of updating both probabilities and learning is successful when $\forall x P(x \in S) = P_R = 1$, that is when the learner can reliably associate the irregular verb with the irregular rule and the irregular rule can reliably be applied over the default rule. Overregularization is predicted to occur when either $P(x \in S) < 1$ (the child cannot reliably associate a verb to its irregular class $S$) or when $P_R < 1$ (the default rule and not the irregular rule $R$ wins the competition). This model also predicts that (a) for two verbs in the same class, the most frequent verb will be used more correctly and (b) for two verbs equally frequent from different classes, the one in the most frequent class will show a more correct use.

In both models, children need to learn which rule(s) are productive and which are not and need to memorize. However, what exactly is memorized is different in each model, as explained above.

3.2 Morphology acquisition in Spanish

Clahsen et al. (2002) analyzed verb inflection produced by 15 Spanish-speaking children taken from longitudinal and cross-sectional samples of spontaneous speech and narratives. They found that regular suffixes were sometimes applied to irregular verbs; in contrast, there were no cases of irregular suffixes applied to regular verbs. They report 168 irregular verb tokens in which there were errors, against 3446 correct irregular forms, yielding an overall error rate of 4.6%. The two most common cases of overregularizations where stem overregularizations (116) ($sabo$ instead of the correct form $se$) and conjugation-internal regularizations (124) ($pusi$ instead of $puse$). In contrast, they found only 2 cases in which errors occurred in regular verb forms (against 2071 correct verbs). Thus, they found a strong difference between regular and irregular verbs in Spanish children’s errors.

Clahsen et al. (2002) did not include the errors regarding diphthongs in these counts. They report 107 tokens in which a non-diphthongized form was produced in a context that required a diphthongized one. There were no cases in which the children produced a diphthongized form in a context that required a non-diphthongized one. They analyzed overregularization rates by grouping verbs according to their sample frequencies. Overregularization rates were higher for verbs with low sample frequencies. They explain these findings through the Words and Rules model, since it postulates the difference between regular and irregular verbs that they found in the data.
3.3 Psycholinguistic and computational studies

Bybee and Pardo (1981) carried out a nonce-probe experiment with Spanish verbs and found an interesting asymmetry between first and third conjugation verbs. When presented with ‘wug’ first conjugation verbs in third person singular, like *bierca* and *duenta*, subjects mostly produced *biercó* (73%) and *duentó* (86%), instead of *bercó* or *dentó*. That is, for first conjugation verbs, subjects assumed that the diphthong was part of the root. In contrast, when subjects were presented with nonce infinitives in third conjugation, subjects mostly introduced an alternation, either diphthongization or raising, depending on the phonological shape of the verb.

Albright et al. (2000) tried to determine whether there are superficial cues that distinguish diphthongizing from non-diphthongizing contexts. Their algorithm derived rules and probabilities for first conjugation verbs. Overall, diphthongization tended to be disfavored in the data, appearing with probability 0.09. Higher probabilities were found for more specific phonological contexts. These probabilities were then tested against Spanish speaker’s intuitions and they found a significant correlation.

4 Data, methods and results

The present study is based on an analysis of the transcriptions of the speech of six monolingual Spanish-speaking children, drawn from the CHILDES database (MacWhinney and Snow 1985): María (1;7-3;10), Magín (1;7-2;7), Irene (0;11-3;2), Juan (2-4), Koki (1;7- 2;11) and Eduard (1;4-3;10).

One of the goals of this paper is to analyze in which circumstances children make morphological mistakes and whether they are correlated with other factors. To that end, all child utterances containing the relevant alternations or failing to present the correct alternation were automatically extracted and manually corrected. That is, we extracted both correct forms, such as *tengo*, and cases of overregularization, in which the child failed to produce the alternation and followed the regular pattern (i.e. *teno*). The following types of verbs were extracted: (1) Verbs with [ie] alternation (IE): *sentar* vs. *siento*, (2) Verbs with [i] alternation (I): *pedir* vs. *pido*, (3) Verbs with [ue] alternation (UE): *lllover* vs. *llueve*, (4) Verbs with velar insertion (VI): *salir* vs. *salgo* and (5) Verbs with velar insertion and ie diphthongization (VI + IE): *tener* vs. *tengo* vs. *tiene*. Additionally, an exhaustive manual search was performed to find cases of overirregularization: i.e. cases in which a child introduced one of the alternations in a regular verb (for example, *cuemo* instead of *como*, first person singular of *comer*, ‘eat’). No such cases were found, in accordance with what Clahsen et al. (2002) reported.

Tables 1.1, 1.2 and 1.3 summarize the data and its main results. Table 1.1 summarizes the errors that each child produced: the number of incorrect tokens (Inc), the number of correct tokens (Corr) and the infinitive of the
verbs in which they produced some errors (the first number in parenthesis indicates the number of tokens of that verb with an error and the second the total count of tokens of that verb that the child produced\(^4\)). The incorrect verbs have been sorted by the type of irregularity they failed to show\(^5\).

<table>
<thead>
<tr>
<th>Child</th>
<th>Inc</th>
<th>Corr</th>
<th>Incorrect verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edu.</td>
<td>7</td>
<td>15</td>
<td>UE colgar (1/1), poder (4/4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VI tener (2/3)</td>
</tr>
<tr>
<td>Mag.</td>
<td>98</td>
<td>618</td>
<td>UE jugar (3/7), poder (3/41)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IE querer (45/365), tener (16/90)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VI tener (10/34), poner (21/43)</td>
</tr>
<tr>
<td>Kok.</td>
<td>134</td>
<td>132</td>
<td>UE colgar (9/9), llover (2/3), poder (14/26), dormir (2/4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IE cerrar (3/5), sentar (15/15), querer (29/86), tener (20/37), venir (13/16)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VI parecer (1/1), poner (15/25), tener (11/31)</td>
</tr>
<tr>
<td>Ire.</td>
<td>44</td>
<td>355</td>
<td>UE contar (2/37), jugar (1/1), sonar (2/2), volar (6/6), llover (2/3), poder (1/6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IE querer (17/77), tener (10/123), venir (1/19), merendar (1/1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VI tener (1/58)</td>
</tr>
<tr>
<td>Jua.</td>
<td>17</td>
<td>73</td>
<td>UE morder (6/8), dormir (3/4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IE querer (3/24), tener (2/10), venir (1/1)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VI tener (2/6)</td>
</tr>
<tr>
<td>Mar.</td>
<td>45</td>
<td>284</td>
<td>UE acordar (1/6), colar (1/1), jugar (5/6), llover (3/5), poder (2/14), poner (1/4), volver (7/7), sonar (2/7), volar (3/4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>IE cerrar (3/8), sentar (1/15), hacer (1/25), querer (3/24), tener (1/21)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VI tener (11/110)</td>
</tr>
</tbody>
</table>

Table 1.1: Error data for every child in the sample

Tables 1.2 and 1.3 summarize the data: the former sorts it by conjugation and the latter by type of irregularity. Each table includes the count of incorrect, correct and total tokens used by children, the correct usage rate (CUR) and the total count of tokens used by adults. The adult counts estimate the input that children received: for each verb with the relevant alternation that the children uttered in the transcriptions, we extracted, from the same transcriptions, the tokens of those verbs that adults uttered. The CUR measures the children’s knowledge of irregular verbs and it is calculated by dividing the number of correct tokens by the total number of tokens, thus it is equivalent to the percentage of correct verbs.

\(^4\)For some of the verbs, children did not make any errors. That’s why the sum of the second numbers in parenthesis is smaller than the number in Corr.

\(^5\)Tokens of verbs with VI+IE irregularity have been sorted depending on whether they failed to show VI (teno instead of tengo) or IE (tene instead of tiene) This is why the verb tener, which shows the two irregularities, appears twice for María.
Laia Mayol

<table>
<thead>
<tr>
<th>Conjugation</th>
<th>Inc</th>
<th>Correct</th>
<th>Total</th>
<th>CUR</th>
<th>Adult Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>60</td>
<td>154</td>
<td>214</td>
<td>0.72</td>
<td>740</td>
</tr>
<tr>
<td>2nd</td>
<td>265</td>
<td>1191</td>
<td>1456</td>
<td>0.82</td>
<td>4711</td>
</tr>
<tr>
<td>3rd</td>
<td>20</td>
<td>132</td>
<td>152</td>
<td>0.87</td>
<td>213</td>
</tr>
</tbody>
</table>

Table 1.2: Data sorted by conjugation

<table>
<thead>
<tr>
<th>Conjugation</th>
<th>Inc</th>
<th>Correct</th>
<th>Total</th>
<th>CUR</th>
<th>Adult Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>0</td>
<td>13</td>
<td>13</td>
<td>1</td>
<td>39</td>
</tr>
<tr>
<td>IE</td>
<td>186</td>
<td>904</td>
<td>1090</td>
<td>0.83</td>
<td>2373</td>
</tr>
<tr>
<td>UE</td>
<td>85</td>
<td>272</td>
<td>357</td>
<td>0.76</td>
<td>1508</td>
</tr>
<tr>
<td>VI</td>
<td>74</td>
<td>288</td>
<td>362</td>
<td>0.80</td>
<td>1744</td>
</tr>
</tbody>
</table>

Table 1.3: Data sorted by type or irregularity

5 Analysis: not all irregularities are equal

5.1 General comments

Tables 1.2 and 1.3 show that the children’s errors are not equally distributed among all classes of verbs. A chi-square test was performed on both tables and confirmed that the differences in both tables are significant (p<0.05 for both distributions)\(^6\). Thus, there is a correlation both between conjugation class and verbs’ CUR and between type of irregularity and verb’s CUR.

As for conjugations, the first conjugation had the lowest global CUR and the third conjugation the highest. The good results for the third conjugation can be easily explained considering that, as mentioned in Section 2, all verbs in this conjugation undergo some alternation, so it is fully predictable that they will undergo either raising or diphthongization. Interestingly, no errors were found regarding raising and the only errors in the third conjugation verbs were in the verbs dormir (‘sleep’, 5 errors) and venir (‘come’, 15 errors), the latter showing both [ie] diphthongization and velar insertion. First and second conjugation verbs are much more unpredictable and, thus, are expected to show lower CUR. However, the counts for both child and adult production for the 2nd conjugation are much higher, since this conjugation includes very common verbs, such as tener (‘to have’), querer (‘to want, to love’) or poner (‘put’). In contrast, first conjugation is the default class and contains more types with fewer tokens.

As for type of irregularity, as mentioned, raising verbs have a global CUR of 1: no child produced any errors on these verbs, in spite of being the class for which the children received the least input from the adults\(^7\).

\(^6\)The test was performed on 3x2 and 4x2 tables, where the row categories were either conjugation class or type of irregularity and the columns categories the incorrect and correct tokens.

\(^7\)As a reviewer notes, there are much fewer tokens for this class and, thus, it would be suitable to have more data. However, the table does reflect the real distribution of verbs.
The irregularity in which the children produced more errors was [ue] diphthongization, while [ie] diphotingization and VI stay in the middle. In these last three classes, there seems to be a correlation between input from adults and CUR.

There are several cases in our data in which the children have a perfect CUR of 1 for a certain verb, despite the fact that it did not occur at all in the input they received from adults in the transcriptions\(^8\). That was the case for the following children and verbs: Irene and apretar (‘press’), María and colgar (‘hang’), Juan and nevar (‘snow’) and servir (‘serve’), Koki and vestir (‘dress’) and Eduard and temblar (‘shake’).

5.2 Statistical analysis

A linear regression analysis was performed on all the verb types which occurred in the overall children speech at least ten times, which was the case for 20 out of the 50 verb types in our data.

The goal of the regression was to predict each verb’s CUR from the following predictor variables: conjugation, type of irregularity and input from adults\(^9\). Best subsets regression was performed and the best model according to Mallow’s Cp statistics (Kleinbaum and Kupper 1978) contained the following two variables: (1) input from adults and (2) whether the verb belongs to the first conjugation or not. With that model, highly significant results were obtained (F=6.1, p = 0.01007).

Figure 1.1 shows a plot of CURs and adult input for every verb included in the analysis. The two lines represent the fitted lines predicted by our model: the higher line corresponds to non-first conjugation verbs and the lower line to first-conjugation verbs. That is, our model predicts that, for a particular verb, the more a child hears the verb, the higher its CUR. Moreover, it also predicts an asymmetry between first conjugation verbs and the rest of verbs, the former having lower CURs. This is consistent with the observations above: third conjugation verbs are predictable, since they always undergo some change, second conjugation verbs are very frequent in the adult and children production. First conjugation verbs have neither of these properties and, thus, children make more errors with these verbs.

5.3 Theoretical implications

The main findings that arise from the data and the statistical analysis are the following:

---

\(^8\)Although this might be due to a sampling effect, the fact that the verb did not occur in the adult speech in our sample means that it occurred rarely.

\(^9\)Since the adult count data did not seem to be normal, it was transformed using a base 2 logarithm. After the transformation, the Shapiro-Wilk test yields a 0.902 probability that the data is normal.
1. Many instances of overregularizations were found (345 tokens out of 1477, global CUR = 0.81). No instances of overirregularizations were found.

2. The two main predictors of a verb’s CUR are (a) adult input and (b) membership to the first conjugation.

3. There are significant differences in the verbs’ CUR according both to the conjugation they belong to and the type of irregularity they present. It is particularly striking that verbs with the raising alternation have a global CUR of 1.

4. Some verbs have a perfect CUR of 1, although there was no adult input for that verb in our data.

While findings (1) and (2a) are compatible with both the Words and Rules and the Rules and Competition model, findings (2b), (3) and (4) pose a challenge to the Words and Rules approach. According to this model, all irregularities are individually stored in memory. Therefore, they predict that overirregularizations should not occur, consistently with what the data is showing. It is also not unexpected that CURs correlate with adult input, since memory storage might be dependent on frequency of exposure. In contrast, it is unexpected under this model that belonging to a class or
showing a particular irregularity should affect the verb’s CUR. Also, it is not expected that children make no mistake at all with irregular verbs rarely seen in the input. The Words and Rules model might be useful to explain past tense acquisition in English since the irregular patterns are not so clear or common (although see McClelland and Patterson (2002) and Yang (2002) for criticism). However, in Spanish, patterns and classes clearly exist within the irregular verbs and, thus, it seems completely inadequate to assume children store each verb individually in their mental lexicon, without taking into account the verb’s conjugation or particular irregularity. A reviewer correctly points out that this data shows that the regular-irregular distinction is not enough and that more fine-grained distinctions are necessary, such as a ‘semi-regular’ category. It is precisely the ‘semi-regular’ verbs or the regular processes within irregular verbs that the Rules and Competition approach can capture, while the Words and Rules cannot.

All the findings are compatible with the Rules and Competition model, which predicts that CURs should be affected by membership to a class and by input frequency. The perfect CUR of third conjugation verbs with a raising alternation is particularly interesting. This finding is fully consistent with Bybee and Pardo’s (1981) study in which subjects mostly introduced an alternation for nonce third conjugation verbs. That is, both children and adults encountering a new verb in the third conjugation seem to be able to reliably determine both the probability that the verb belongs to a particular class (i.e. $P(x \in S) = 1$) and that a particular irregular rule needs to apply ($P_R = 1$), possibly depending on the phonological form of the verb. I leave for future work to test whether the two specific predictions of the Rules and Competition model hold for the data I presented. However, since classes within irregularities play such an important role in this model, it is a much more promising way of thinking about how children can learn the persistent irregularities they encounter in Spanish verbal morphology.

6 Conclusion

This paper has analyzed data from acquisition of irregular patterns in Spanish verbal morphology. Three-hundred and forty-five tokens of overregularization in child speech were gathered and a statistical analysis was performed. The statistical analysis revealed significant differences in the verbs’ Correct Usage Rate depending on conjugation and type of irregularity. Moreover, the best statistical model proposed adult input and first conjugation membership as predictors for a verb’s CUR. The data presented here presents challenges for the Words and Rules model, while it is consistent with a model that takes into account classes to explain irregularities, such as the Rules and Competition model.
Bibliography


Acquisition of Irregular Patterns in Spanish Verbal Morphology
An Approximate Gazetteer for GATE based on Levenshtein Distance

BRUNO WOLTZENLOGEL PAULO
Technische Universitaet Wien
bruno.wp@gmail.com

Abstract. GATE is an architecture that allows the use and development of useful plugins for typical Natural Language Processing tasks. However, there is currently no plugin capable of annotating a document that contains words that only approximately match words specified in a list of words to be searched. Here we describe the development of such a plugin, based on Levenshtein Edit Distance, and its integration to the GATE environment. This plugin enables GATE to be useful for tasks in which exact matching is not enough.

1 Introduction

GATE (General Architecture for Text Engineering) provides a simple abstraction of a typical Natural Language Processing task ([3],[2]). In this abstraction, each task is defined as a pipeline, consecutively applying different processing resources (Part-of-Speech taggers, Sentence Splitters, Gazetteers, ...) to a given language resource (a text document or a collection of text documents). The output is a language resource enriched with annotations.

Here we are interested in a particular type of processing resource called Gazetteer. In section 2 we explain what is the purpose of a gazetteer, what annotations it produces and how the DefaultGazetteer provided by GATE works. In Section 3 we briefly describe a few applications in which the DefaultGazetteer is not useful, because it is not able to cope with noise or errors in the language resource. These applications are the motivation for the implementation of a new gazetteer, here called AproximateGazetter. Section 4 describes its ideas and algorithms and, finally, in Section 5, we evaluate this new gazetteer.

2 Gazetteers

Gazetteers search for words in texts. Whenever one word belonging to one of the gazetteer’s lists of words is found in the text, the matching region
An Approximate Gazetteer for GATE based on Levenshtein Distance

in the text is annotated with the majorType and minorType associated with the list to which the word belongs. If, for example, one of the lists of the gazetteer is a list of city names, then for each matching city name encountered in the text, the gazetteer will produce an annotation with the following parameters:

- **Start** - the position where the matching word starts.
- **End** - the position where the matching word ends.
- **Features** - a set of features associated with the annotation (for the city example, it would be majorType=location, minorType=city)
- **Lookup** - the set of annotations to which this annotation belongs.

### 2.1 GATE’s DefaultGazetteer

Gate provides, within its plugin ANNIE (A Nearly New Information Extraction System) ([1]), the interface **Gazetteer** and the class **AbstractGazetteer**, which specify the minimal requirements of a gazetteer in terms of properties and methods that they should have and implement the basically functionality of some methods. Additionally, it provides the class DefaultGazetteer, extending **AbstractGazetteer** and implementing Gazetteer into a fully functional gazetteer ([6]).

**DefaultGazetteer** is based on finite state machines (FSM). When it is loaded, all the words in its lists are read and a FSM is built combining all of them so that their characters are the labels in the transitions between the states, and a state is final whenever it corresponds to the end of a word. When the **DefaultGazetteer** is executed on a text, it reads the characters of the text and performs the corresponding transitions in the FSM, adding annotations when it finds final states.

The lists of words to be searched and annotated are simple plaintext files with one word in each line, and the gazetteer becomes aware of these lists by reading another file (“lists.def”), in which each line points to a different list that should be used and specify its majorType and minorType features. An example of “lists.def” might look like:

```
city.lst:location:city
country.lst:location:country
```

In this case, the gazetteer will search all words specified in the plain-text files “city.lst” and “country.lst” and will annotate the occurrences found with the features majorType=location, minorType=city for words in “city.lst” and majorType=location, minorType=country for words in “country.lst”.

198
3 Applications of Approximate String Matching

Although the DefaultGazetteer, based on FSM as described previously, is very efficient and has a good precision, its recall may be not so good for some applications, because it is only able to detect exact matches of the words. If one character is mistyped in the name of a city in the text, for example, this occurrence will not be found. The application areas briefly described below are examples that require a more flexible Gazetteer, capable of detecting occurrences with noise and errors:

- **Bioinformatics** - one of its common tasks is to align sequences of DNA or Aminoacids in genes or proteins. These sequences can be seen as texts and since genes and proteins may contain mutations and abnormalities, approximate (inexact) matching is necessary for the alignment.

- **Information Retrieval** - if the documents that are being analyzed contain errors and noise, the demanded document might not be found simply because the relevant word contained errors and hence it couldn’t be found by exact matching. Noise and errors in documents can come from digitalization via Optical Character Recognition (OCR) or from Speech recognition technologies.

- **Text Classification** - Text classification based on the annotations of a gazetteer may have their accuracy hindered if the gazetteer does not find noisy occurrences of the relevant words.

- **Multi-language Search** - Some languages are similar in their vocabulary, having words with the same root but with slightly different derivational morphology (e.g. “Algeria, Algrie”, “Andorra, Andorre”, “Bhoutan, Bhutan, Butao”). Different morphology may also occur as a result of internet slang. A gazetteer with approximate matching would be able to recall all these words, without knowledge of the morphology particularities of each language.

- **Text Correction and Completion** - A system might suggest corrections and completions for a word that has matched only approximately some word in the list of words. This may be useful and feasible for applications with a limited vocabulary.

- **Recovering from Noise in Signal Analysis** - By finding the best match for a noisy signal according to an allowed code, it is possible to recover the intended message.

By programming a simple approximate gazetteer, here called *Aproxi-mateGazetter*, we expect to expand the frontiers of application of GATE
An Approximate Gazetteer for GATE based on Levenshtein Distance
to all these areas. We also note that an improved recall provided by an app-
proximate gazetteer may also benefit already classical areas of application
of GATE, since other types of processing resources usually use the results
of gazetteers. Sentence-splitters, for example, would benefit from a better
detection of abbreviations by a gazetteer.

4 The Classic Levenshtein Distance and Some Improvements

To determine whether a string approximately matches another string, we
need a distance function to measure how distant from each other the two
strings are. There are different such functions ([8],[5],[4]). For the Approx-
imateGazette, we chose the Levenshtein distance [7], because it seems
to be a good compromise between flexibility to deal with different kinds of
errors and efficiency of the algorithms that compute it.

The levenshtein distance between two strings is defined as the minimum
number of operations to transform one string into the other, where the oper-
ations may be deletion of a character, insertion of a character or substitution
of a character. Thus the distance between “aaa” and “aaaa” is 1, because
one ‘a’ must be inserted in the first string to make it equal to the second
string. Analogously, the distance between “aba” and “aca” is also 1, because
‘b’ must be substituted by ‘c’. The different operations may have costs that
are different from 1 and the cost may even depend on the characters that are
inserted, deleted or substituted. This allows us to particularize the distance
to specific types of errors and noise.

To compute the minimum distance, a dynamic programming algorithm
can be used. We incrementally fill a bi-dimensional array $D$, where $D[i][j]$ stores the distance between the initial prefix subtrings of length $i$ and $j$ respectively of the first and second string. Clearly, to compute $D[i][j]$, we just need to take $D[i−1][j−1]$, if the characters in position $i$ of the first string and in position $j$ of the second string are equal, or we need to take the minimum of $D[i−1][j−1]+\text{substitutionCost}$, $D[i][j−1]+\text{insertionCost}$, $D[i−1][j]+\text{deletionCost}$. Thus we can fill the array from left to right and from top to bottom, since in this way we guarantee that we always have previously calculated the values that we need to calculate $D[i][j]$. Once the array is finished, we can read the final distance between the two strings in its bottom-right corner. If this distance is below a fixed threshold, we may be willing to consider both strings to match each other. Table 4 shows an example of such a computation.

This algorithm is $O(n.m)$ both in time and space, where $n$ and $m$ are
the lengths of the strings.

200
Table 1.1: Example of computation of Levenshtein distance between the strings “CATO” and “GATE”.

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>T</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>G</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>T</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

4.1 Finding Several Matches in a Text

The algorithm presented previously decides whether 2 strings match, but it’s not able to decide whether 1 of the strings matches a substring of the other string (which may be a much longer string, containing a whole text full of words and sentences). Neither it is able to find the positions where these matches occur.

A very naïve approach to solve this problem would be to compute the distance of the string to each possible substring of the text. However, this would be too slow, because a text of length m would have \( m^2 + 2m + 1 \) possible substrings. A slightly less naïve approach would be to consider only some of these substring, only those separated by spaces and presumed to be words, for example. But with this approach, we cannot handle errors related to insertion of spaces in the middle of words or removal of spaces between words.

Fortunately there is another solution which is as computationally expensive as solving the problem of simply computing the levenshtein distance between two strings [8]. Let’s assume that the text (the bigger string) is horizontal in the distances array (i.e. the array has as many columns as characters in the text plus one). Then by initializing the first row of the array with zeros, we implicitly tell the algorithm that the matching of the string may start at any position in the text. And then we can detect the matches by looking at the last row and seeing which cells contain values that are below the distance threshold.

However this solution (as described in [8]) only finds the end position of a match. To determine its start position, we may compute for each cell of the array not only the distance but also the number of deletions and insertions that were used to yield that distance. And then we can find the start position according to:

\[
\text{startposition} = \text{endposition} - (\text{lengthOfString} + \#\text{insertions} - \#\text{deletions})
\]
Table 1.2: Example of detecting matches in a text with the dynamic programming algorithm. Searching for “CA” in “CATACA” with distance threshold equal to 0. End-positions of the matches are marked by the ‘0’s in the last row, which are emphasized in italic bold.

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>T</th>
<th>A</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

For example, searching for the word “York” in the text “New York” with distance threshold equal to 1, will first give us two matches, one of them with endposition in the 7th character and 1 deletion operation and the other one in the 8th character with 0 deletion operations. Then we compute the startposition for both of them and the result is the 5th character.

This algorithm, although sound, is still incomplete. It didn’t find the matching of “York” with the substring “ork” in the text, which requires only 1 deletion of the character ‘Y’. In order to fix this and obtain a complete algorithm, all we have to do is to consider more matches (and add the corresponding annotations), with increasing values of startposition, until the distance threshold is not satisfied anymore. For the above example, after we computed the startposition to be the 5th character for the match of “York” with “York” with distance 0, we consider increasing the startposition to the 6th character. This corresponds to the deletion of a character and thus the distance of this candidate match would be 1. Since this distance is below the threshold, we annotate this match. Then we consider increasing the startposition to the 7th character. This corresponds to 2 deletions and thus to a distance of 2, which is above the threshold. Therefore we don’t annotate this candidate match and we stop increasing the startposition.

### 4.2 Avoiding Overlapping Matches

The procedure described above generates several overlapping matches (and their corresponding annotations) on an occurrence of the word, each match with different degree of error within the distance threshold. In most applications, these overlapping annotations are not desirable. The algorithm should return only the best (minimum distance) of such overlapping matches.

To find only the endposition of the best matches, it suffices to analyze the last row of the table, considering the sequences in which the distance is below the threshold. For these sequences, the best (non-overlapping) matches are only those that are local minima of distance. Table 4.2 illustrates the avoidance of overlapping matches.
Table 1.3: Avoiding overlapping matches with a distance threshold of 2. Normally, we would obtain 4 matches (shown in italics in the last row), but in order to avoid overlapping, our algorithm may return only those corresponding to local minima (shown in bold in the last row) in the sequences of matches

<table>
<thead>
<tr>
<th>T</th>
<th>C</th>
<th>A</th>
<th>T</th>
<th>G</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>T</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Since, in general, an application may need the overlapping annotations, our implementation allows the user to choose whether he wants to avoid them or not.

4.3 Saving Memory Space

For very long texts, the size of the distance array can be quite big. And it doesn’t need to be, if we note that to compute a column, we just need the previous column. Hence, the memory use of the algorithm may be reduced from $O(n.m)$ to $O(n)$. The time complexity continues to be $O(n.m)$.

4.4 Normalization of Distances

Since the size of strings vary, it is not very convenient to deal with absolute distances. One character mismatch in a short word is much more relevant than one character mismatch in a long word. Hence our ApproximateGazetteer deals instead with relative distances. The relative distance threshold is specified in a range from 0.0 to 1.0 and then for each word that is searched, an absolute distance threshold is computed by multiplying the relative threshold by the length of the string. Then the algorithm is applied normally with this absolute distance threshold.

5 Evaluation of the ApproximateGazetteer

To evaluate our algorithm, we were primarily concerned with its time performance and its accuracy, in comparison with GATE’s DefaultGazetteer.
5.1 Time Performance

It is very clear that \textit{ApproximateGazetter} is qualitatively much slower than \textit{DefaultGazetteer}. This happens because \textit{ApproximateGazetter} has to scan the text once for each word that is searched for, while \textit{DefaultGazetteer} compresses all words in a FSM and thus is able to search for all of them simultaneously.

This linearity of \textit{ApproximateGazetter}’s time performance with respect to the number of words that are searched for, constitutes its main disadvantage and restricts its practical usage to small lists of words. This was the price to pay for its extra functionality.

5.2 Accuracy

An interesting way to statistically evaluate the accuracy of \textit{ApproximateGazetter} would be to measure its recall and precision for a large annotated corpus of noisy text. However we did not have access to such a large corpus and therefore we evaluated our algorithm only in a small domain-specific example, in order to find possible directions for future works and improvements. As language resource to be processed, we used a single XML file (“data.xml”) containing several names of cities and their geographical coordinates. A short part of the file can be seen below:

```xml
<?xml version='1.0' encoding='ISO-8859-1'?>
<data>
  <countries>
    <country>
      <name>Argentina</name>
      <cities>
        <city>
          <name>Bariloche</name>
          <latitude>-41.150</latitude>
          <longitude>-71.300</longitude>
        </city>
        <city>
          <name>Buenos Aires</name>
          <latitude>-34.613</latitude>
          <longitude>-58.470</longitude>
        </city>
      </cities>
    </country>
  </countries>
</data>
```

We processed this file with \textit{ApproximateGazetter} (with “avoidOverlappingAnnotations” set to true and “normalizedDistanceThreshold” set to
0.15) and with GATE’s DefaultGazetteer. Then we compared the annotation sets produced, by using GATE’s Annotation Diff Tool. The summarized output of the tool is displayed in tables 5.2 and 5.2:

It is important to note that this test takes the annotations produced by AproximateGazetter as standard (“key”) and measures how well the annotations by DefaultGazetteer (“response”) agree with the key. We can see that AproximateGazetter produced the same annotations produced by DefaultGazetteer (“Correct Matches”=265 and “Partially Correct matches”=1) plus some more (“Missing” = 99). By analyzing the details of the annotations that DefaultGazetteer missed, we can see that:

- Many of the misses were due to the fact that DefaultGazetteer is not able to produce annotations within annotations. The city “San”, for example, was not annotated when it occurred inside a larger city name (“San Pedro de Atacama”, “Santiago”). Although there must be reasons for this behaviour of DefaultGazetteer, there may be situations where the inner annotations are also important and shouldn’t be ignored. AproximateGazetter does not ignore them.

- AproximateGazetter was able to recognize words containing characters with diacritics, which were not detected by DefaultGazetteer: “Münster”, “Düsseldorf”, “São Paulo”.

- it recognized words with extra spaces: “Aguas Calientes” matched “Aguascalientes”.

- it handled capital letters well without preprocessing the text to UpperCap: “Aix-en-Provence” matched “Aix-En-Provence”.

- it recognized words in different but similar languages: “Sevilla” matched “Seville”, “Barcelona” matched “Barcelone”, “Brussel” matched “Brus-
An Approximate Gazetteer for GATE based on Levenshtein Distance

“sels”, “Kopenhagen” matched “Copenhagen” and “Copenhage”, “Granada” matched “Grenada”, “Cordoba” matched “Cordova”, “Iraklio” matched “Iraklion”, “Korinth” matched “Corinth”, “Hannover” matched “Hanover”.

• it had a few false matches: “aracas” (from “Paracas”) matched “Cara
cas”, “Argentina” matched “Argentia”, “othenburg” (from “Rothen
dburg”) matched “Gothenburg”.

6 Conclusions

The goal of this project, to implement an approximate gazetteer for GATE, was successfully achieved and the source code may be downloaded from [9]. Additionally, some theoretical modifications and improvements of Levenshtein’s Edit Distance were discussed. One of its possible uses was qualitatively demonstrated by processing a resource with names of cities in possibly different languages. AproximateGazetter opens a new area of application for GATE, which is that of processing language resources with noise and errors. However, it is just a first step and much future work has to be done. Among the possible directions for future work, we would like to mention:

• Statistically test the algorithm with large degraded and noisy corpus.

• Find ways of improving the speed of the algorithm, possibly trying to combine ideas of FSM into the dynamic programming algorithms. We note, for example, that for two words with the same prefix of length n, the first n rows of the dynamic table are identical. Therefore we could try to devise a better data structure to allow us to save this kind of repeated computations for prefixes.

• Extend our algorithm to deal with ontologies, as the DefaultGazetteer was extended to the OntoGazetteer.

• Give more options to the user. For example, how spaces in words or between words should be handled or whether subwords in the text can be matched to the words in the lists of the gazetteer.

• Experiment with other string distances, different from Levenshtein’s.

Bibliography


An Approximate Gazetteer for GATE based on Levenshtein Distance
Obligatory adjuncts in weak accomplishments

MÁRTA PEREDY
Research Institute for Linguistics, Hungarian Academy of Sciences
mperedy@nytud.hu

ABSTRACT. The primary goal of this paper is to provide a unified explanation for the Obligatory Adjunct Phenomenon in English and in Hungarian. My explanation is based on the assumption that accomplishments fall into two types with different syntactic and semantic properties, called weak and strong accomplishments, both in Hungarian and in English. The infelicity of the sentence The house was built results from the fact that it is grammatical in contexts where it can be interpreted as a strong accomplishment and it is ungrammatical if it can be interpreted only as a weak accomplishment. I claim that the so-called Obligatory Adjunct Phenomenon (OAP) can be considered as a subcase of the neutralization of the requirement that weak accomplishments have a non-specific theme argument.

1 Introduction

English passive sentences consisting of a verb of creation and a definite theme are infelicitous (acceptable only in certain contexts, as indicated by the #), while the appearance of an adjunct makes these sentences grammatical (see (1a-1b)). In these cases the adjuncts of the grammatical sentences are traditionally called obligatory adjuncts. The phenomenon was first discussed by Grimshaw and Vikner ([8]), who also noticed that there is no need of an obligatory adjunct if the theme argument is non-specific (1c), though they could not integrate this fact to their account.

(1) a. #The house was built.
   b. The house was built in Budapest.
   c. A house was built.

The Hungarian counterpart of (1) is (2). The phenomenon is parallel in the two languages. Beside some language specific differences (e.g. Hungarian uses unaccusative constructions instead of the passive), there is one important difference between (1a) and (2a). While (1a) is said to be only infelicitous, (2a) is clearly ungrammatical (in the intended perfective reading). The accent indicates the position of the main stress of the Hungarian sentence.
Obligatory adjuncts in weak accomplishments

(2) a. *Épült a ház.
   built-unacc-PERF the house
   'The house was built.'

b. 'Budapesten épült a ház.
   Budapest-in built-unacc-PERF the house
   'The house was built in Budapest.'

c. 'Épült egy ház.
   built-unacc-PERF a house
   'A house was built'

The judgement of the grammaticality of the English sentence is far from being clear. In a context where the concept of the house is already a part of the discourse, (1a) can be used (see (3a)), while according to the Guess what? test of G&V, this can’t be uttered just out of the blue (see (3b)).

(3) a. John had always wanted to build a house, and after winning the lottery, the house was built.

b. Guess what? *The house was built.

The appropriate Hungarian translation in the (3a) context is not (2a), but contains the prefixed version of the bare verb of creation:

(2a') Megépült a ház.
   PREF-built-unacc-PERF the house

In what follows, I will show that English sentences like (1a) are not infelicitous but ambiguous: there is a grammatical and an ungrammatical reading (like in Hungarian: (2a') and (2a)). In order to maintain this parallelism between the two languages, I will argue that it is reasonable to distinguish the so-called weak and strong accomplishments in English, too. On the basis of this distinction, I will suggest an explanation for the Obligatory Adjunct Phenomenon (OAP), covering both the English and the Hungarian data. There is a need of such a unified account since the two previous accounts are not applicable to Hungarian in their present form.

2 Previous accounts of obligatory adjuncts

2.1 Grimshaw and Vikner (1993)

Considering the example in (4) beside the examples in (1), Grimshaw and Vikner (G&V) claim that it is only a well identifiable class of verbs that require obligatory adjuncts in passives: the verbs of creation. (Verbs like build, draw (a picture), dig (a hole) are members of this class, but destroy for example is not.)

(4) The house was destroyed.
They argue convincingly against the most obvious idea that obligatory adjuncts would be arguments of the verb. These adverbials are not assigned particular thematic roles:

(5) The house was built *yesterday / by John / of wood.*

G&V suggest (among other authors) that each subevent of the complex event must be identified by some element in the sentence. In the case of creative accomplishments, the theme argument refers to the final state, while the agent refers to the process. In the short passive, however, there is no element identifying the process since the agent is absent. Obligatory adjuncts make these sentences acceptable by being able to identify the process.

Though the basic idea that event structure has its role in the phenomenon turns out to be fruitful in the present analysis, the following facts do not support the previous reasoning:

1. In Hungarian the presence of the agent does not render the sentence acceptable:

(6) a. *János építette a házat.*
   John built-PERF the house-ACC

b. John built the house.

2. Sentences which contain a non-specific theme do not require any further constituent (obligatory adjunct) (see (1c, 2c)).

3. The presence of the adjunct can not render the sentence acceptable in itself, but it should occupy the position of the information focus, i.e., carry the main stress of the sentence. The natural position of the focus is at the end of the English sentence and it is the immediately preverbal position of the Hungarian sentence. (Cf. (1b, 2b) and (7, 8))

(7) # In September, this house was built.

(8) *A ház épült szeptemberben.*
   the house built_{unacc-PERF} September-in

2.2 Goldberg and Ackerman (2001)

Contrary to the event structure account, Goldberg and Ackerman’s account (G&A) is based on the main assumptions of conversational pragmatics. They claim that sentences like (1a) are problematic because the Focus Requirement is not satisfied. This requirement, claiming that every utterance must have a focus that serves to convey new information in the discourse, is derived from Grice’s Maxim of Quantity.
In the sentence *The house was built* the definite noun phrase presupposes the existence of its referent, while the creation event which is the predicate of the sentence asserts the existence of the same entity. These two things lead to a contradiction, and there is no relevant information conveyed by the sentence.

G&A emphasize that obligatory adjuncts are only one of the possible ways of satisfying the requirement.

This account provides an explanation to the information structural facts that remained unexplained in the previous section, but the contrast between the two languages in the active case (6) is not clear. They can not give the exact conditions of the acceptability of the infelicitous case (1a) either. How can it happen, that in certain contexts the sentence is still informative, if the reasoning above stands?

In Hungarian, the informative and uninformative cases distinguish overtly (see (2a) and (2a')), in the grammatical case a verbal particle appears in the sentence.

(9) # The house was built.
   a. * 'Épült a ház. (see (2a))
   b. 'Megépült a ház. (see (2a'))

In the next section, I will discuss the relation of these two Hungarian sentences and the relation of both to the OAP. In section 4, the parallel interpretation of the English data will be presented.

### 3 The Hungarian obligatory adjunct phenomenon

The following two questions should be answered in this section:

What is the difference between the two events represented by the Hungarian sentences (9a) and (9b)? What is the reason why there is no need of an additional focused element (an obligatory adjunct) in the case of the prefixed verb?

The grammatical version of (9a) contains a non-specific theme:

(9) c. 'Épült egy ház. (see (2c))

#### 3.1 Descriptive differences

(9a) and (9c) contain a bare verb of creation, (9b) contains a prefixed version of it. The prefixed version consists of a verb of creation and a verbal particle (*meg*) immediately preceding the verb. This particle carries the main stress of the sentence, while in the bare case the verb does. The intended reading is perfective in both sentences. The verbal particle *meg* expresses the culmination (or finishing) of the telic event.
3.2 Definiteness Effect

The Definiteness Effect was first described in the context of existential sentences \( \text{(there is/are} \ldots \text{)} \) ([12]), but later Szabolcsi ([17]) broadened the notion to a large class of verbs, and called them DE-verbs. The theme argument of the DE-verbs must be realized by a non-specific noun phrase on a perfective reading in a neutral clause (i.e., when the main assertion is the event expressed by the verb, cf. (9c)). The common feature of the DE-verbs is the EXIST meaning component which expresses the existence, the coming into existence or the causation of coming into existence of the theme but in a broader sense, i.e., these verbs introduce new discourse referents realized by the theme argument into the discourse. Verbs which fall in the scope of the OAP turn out to be DE-verbs (e.g., (9a)). Another restriction called Specificity Effect holds to the prefixed counterparts of the DE-verbs; they require a specific theme (cf. (9b)) ([3]).

The Definiteness Effect was discussed by several authors in the literature on Hungarian ([3], [9], [11], [17]). The core of their explanations is the same as that of G&A: in the case of a DE-verb and a specific theme argument, the assertion and the presupposition of the existence of the same discourse referent leads to contradiction.

3.3 Culmination of accomplishments

Another way to grab the difference is the type of the culmination of the telic events in (9b) and (9c). According to [4], [15] and [10], there are two possibilities in Hungarian:

1. The culmination is expressed with an oblique argument or a verbal particle which refer either to the terminus or to the result state. For example, in (9b) the verbal particle \textit{meg} refers to the result state.

2. The culmination is incorporated into the internal argument. “The descriptive content of the [internal argument] does not necessarily identify what the theme is like throughout the process, but it refers to what it becomes as a result of the culmination.” (K&V) Thus, in neutral sentences the theme can not be discourse-linked, i.e., specific. (9a) and (9c) show this.

3.4 Intentional entities

The requirement on the (non-)specificity of the theme and the culmination type are not independent notions. They can be best understood on the basis of the \textit{intentionality} of the theme. In the case of incorporated culmination (see 3.3), although the created object (e.g., the house as a physical entity) does not exist until the culmination of the event, the idea of the object or

(Just for the sake of completeness, I mention that (9a) also has an imperfective reading which is grammatical.)
Obligatory adjuncts in weak accomplishments

the intention of creating the object may exist (e.g., the plan of the house) ([13], [10], [1]). Intentional entities refer to the idea of the actual object to be created.

To clarify the relation of the previous four properties the following holds:

1. The culmination of events expressed by DE-verbs is incorporated into the theme argument. It follows from this that the theme refers not to an intentional entity, but to the actual entity to be created, and therefore in neutral sentences it must be non-specific.

2. The culmination of events expressed by prefixed DE-verbs is represented by the verbal particle which refers to the finishing of the creation event expressed by the DE-verb. The theme is an intentional entity the actualization of which comes into existence by the end of the creation event. The theme argument is specific as intentional entities can not be referred to non-specifically.

Example (10) demonstrates the difference between the two event types. While the user of (10a) may have brought a big box suggesting that we use it as a table in the lack of a real one, (10b) could not be used in the same context. (10b) presupposes the existence of (the idea of) a real table.

(10) a. 'Hoztam egy asztalt.
   brought-I-PERF a table-ACC
   'I’ve brought a table.'

b. 'Meghoztam egy asztalt.
   PREF-brought-I-PERF a table-ACC
   'I’ve brought one of the tables.' (Example from Szabolcsi)

3.5 Weak and strong accomplishments

A further property shows the event structure difference as well. Events expressed by DE-verb have simple event structure, while events expressed by prefixed DE-verbs have complex event structure ([4]). Piñón ([15]) grabs this difference in general by distinguishing two subclasses of the accomplishments with the aid of the almost-test:

- *strong accomplishments* are compatible with in-adverbials and have both a counterfactual and a scalar interpretation with almost.

- *weak accomplishments* are compatible with in-adverbials but have only the counterfactual interpretation with almost.

(11a) shows a strong accomplishment, while (11b) a weak accomplishment.

(11) a. Majdnem megépült a ház.
   almost PREF-built\textsubscript{unacc}-PERF the house
counterfactual: the building of the house did not begin
scalar: the building of the house is not finished

b. Majdnem épült egy ház.
almost built^{unacc-PERF} the house
counterfactual: no building of any house began
no scalar reading

The verbal particle of strong accomplishments encodes a notion of finishing that weak accomplishments do not encode. In the case of the scalar reading, it is only this finishing meaning component that is negated.

Weak accomplishments are exactly those events which are expressed by DE-verbs (also called creation events), while the complex events expressed by the prefixed DE-verbs are strong accomplishments.

3.6 Neutralisation of the DE

On the basis of the previous sections it is clear that the constructions which require obligatory adjuncts (more precisely: focus) are weak accomplishments. These, however, require a non-specific theme in neutral sentences. One question remained: Why is a specific theme licensed in the presence of a focused constituent?

The phenomenon that in non-neutral sentences the theme argument of the DE-verb may be specific is well known in the literature and is called the Neutralisation of the DE. If the creation event belongs to the presupposed part of the sentence, for example due to a focused constituent, the theme, the created object is also presupposed, i.e., specific. ([17], [9], [3], [11])

Obligatory adjuncts are focused constituents in weak accomplishments, the requirement on obligatory adjuncts is actually the same phenomenon as the well known neutralisation of the DE.

4 Weak and strong accomplishments in English

While in Hungarian the existence of verbal particles makes the difference transparent (e.g., épül, megépül), in English the same verb form (build) expresses both types. It follows from the fact that, according to É. Kiss ([5]), in Hungarian it is the event structure that is grammaticalized, while in English it is rather the viewpoint aspect.

In the previous section we have seen that in Hungarian strong accomplishments do not require obligatory adjuncts at all, and weak accomplishments require them if their theme is specific. The aim of this section is to show that the same holds in English.

It is important to distinguish the accessibility of a certain reading from its grammaticality. Accessibility means that the lexical semantic (and syntac-
Obligatory adjuncts in weak accomplishments

tic) selectional criteria of the verb are satisfied. Hence, the ungrammaticality of an accessible reading can follow only from the information structure.

The source of infelicity of the sentence The house was built. lies in that both readings are accessible, but the strong reading is grammatical, while the weak reading is not. If both readings are grammatical or if the grammatical reading is strengthened by some factors while the ungrammatical one is blocked, the resulting sentence is acceptable without any difficulties.

4.1 Tests of the accomplishment types

In this section, I will discuss the points of reference on the basis of what the accomplishment type of neutral sentences can be decided in English. In section 4.2, the sentences with focus will be analysed.

Intentionality of the theme

The theme of the weak accomplishment must refer to an actual object, while the theme of the strong version must refer to an intentional entity. Referential noun phrases are usually ambiguous, so this distinction can not be used as a test. But if it is once made clear which reading is the intended, then the intentionality of the NP decides whether the event described by the English DE-verb is a weak or a strong accomplishment.

(12) The house, the plan of which was refused by the municipality, was built.

(13) * This ugly house in front of ours was built.

(In the examples below ((15-25)) the phrase this house here will refer to the actual object reading. If this is starred, that will indicate the ungrammaticality of the weak accomplishment reading.)

Specificity

One case when the noun phrase of the theme obviously reveals the event type is the case of the non-specific theme. Due to the fact that intentional entities are always specific, the strong accomplishment reading is not accessible. The specificity of the indefinite is transparent neither in English nor in Hungarian. The which-test, however, is an indicator of specificity. If the intended reading is non-specific in (14a), the question in (14b) is infelicitous.

(14) a. A house was built. / John built a house.

b. # Which house?

Culmination

In Hungarian, the verbal particle expresses the culmination of the strong accomplishment, but in English it is not represented by an overt element. Supplementary factors can indicate the culmination.

216
(15) The house/*This house here has been built.

Perfect tenses focus to the result state of the event which implies the culmination. As the specific theme is topicalized in the passive sentences, this can not interpreted as the main assertion and thus the culmination of the sentence. So the weak accomplishment interpretation is blocked and the strong is strengthened (15).

The presence of the agent in the active sentence also implies the successful culmination of the telic event unless the culmination is excluded by the use of the progressive aspect. If the theme of the neutral sentence is specific, it can not refer to the culmination, i.e., again the weak accomplishment interpretation is blocked and the strong is strengthened (16).

(16) John built the house/*this house here.

The actual world

In this section, we returned to the problem of the intentionality of the specific theme. There are some cases where the actual object interpretation is blocked as it is not present in the actual world at the utterance time, and therefore can not be referred to by a specific NP. The future tense, the progressive aspect, the use of modals or the negation – all lead to this effect.

(17) The house/*This house here will be built.
(18) The house/*This house here was being built.
(19) The house/*This house here may be built.
(20) The house/*This house here was not built.

4.2 Sentences with focus

The class of sentences with focus divides to two subclasses. In one case the function of the focus is to strengthen the strong reading and to block the weak one while in the other case both readings are grammatical.

When the focus disambiguates the sentence

In the following three cases it will turn out that the theme argument must have intentional reading. In the case of the verum focus (21), where the truth value of the sentence contrasts with its negated counterpart, and in the case of the focused theme (22), which contrasts with an object which was not built, there is a contrast between the sentence and a negated alternative. As we have seen above (20), in the negated sentence only the intentional interpretation is possible. Thus, assuming that the theme of the alternative and the theme of the sentence should have the same intentionality, the intentional nature of the theme follows.

(21) The house/*This house here WAS built.
Obligatory adjuncts in weak accomplishments

(22) The HOUSE/*This HOUSE here was built, not the GARAGE.

In the third case (23), the contrast is established between the final stage and a previous stage of the creation by the focusing of the verb. The physical object doesn’t have previous stages before its coming into existence at the final stage, so the specific noun phrase can refer only to the intentional entity.

(23) The house/*This house here was BUILT, not just DESIGNED.

Foci license both weak and strong accomplishments

The stressed verb may awake also another set of alternatives, where there is a contrast between the “in the specific way” meaning component of the verbs (24).

Finally, I return to our starting point: (25) shows the case of focused adjuncts (or focused arguments different from the theme). In (24) and (25), the weak accomplishment reading is also grammatical, as the focus leads to the presupposition of the creation event itself and licenses the specific actual object. That is why focused constituents seem to be obligatory.

(24) The house/This house here was BUILT, not BOUGHT.
(25) The house/This house here was built of WOOD.

5 Conclusion

On the one hand I agreed with G&A on that obligatory adjuncts serve to convey the new information of the sentence. On the other hand, it was shown on the basis of Hungarian that event structural facts also play an important role. I claimed that English verbs in question are ambiguous; it is actually only one of the two possible readings that fall under the scope of the OAP (the so-called weak accomplishment reading (WAR)).

Three major conclusions:

First, in the WAR the verbs select an actual entity as their theme, which expresses the culmination of the event, while in the so-called strong accomplishment reading they select an intentional entity. In this case the culmination is expressed overtly by a verbal particle in Hungarian, and implicitly (e.g., by means of aspect) in English.

Second, in the end, the OAP turns out to rely on a selectional criterion of the verb, but this applies not to the adjuncts but to the theme.

Third, those neutral sentences require obligatory adjuncts in which the WAR is the intended, though the theme is specific. Obligatory adjuncts, beside other possibilities, can serve as a focus, so by presupposing the weak accomplishment, license the specific theme argument.
Bibliography


[13] Piñón, Ch. 2006a. Verbs of Creation in Hungarian. 7th ICSH, Veszprém


Obligatory adjuncts in weak accomplishments


Guessing Text Type by Structure

Olga Pustylnikov
University Bielefeld
olga.pustylnikov@uni-bielefeld.de

Abstract. Do structural patterns tell us something about the text content? In order to give a preliminary answer to this question, an experiment in automatic text classification is performed using only surface-structural features. The features extracted – as, e.g., the complexity, length or depth of text constituents (Köhler, 1999) – produced surprisingly good classification results. This finding argues for a deeper exploration of structure as an indicator not only of text content, but possibly also of the functional properties of language.

1 Introduction

The search for appropriate features of text classification is a central task in information retrieval and in other relevant disciplines. The goal in this area is to gain as much information about the text by processing as little information from the text as possible. The commonly used methods to address this problem are based on the vector space model (Baeza-Yates and Ribeiro-Neto, 1999) representing texts as a bag of features and comparing them by means of their feature values. In most cases the features used correspond to the frequencies of the lexical items occurring in the texts. From a linguistic point of view such an approach, even though it performs well in most cases, focuses solely on similarities between words, but does neither take the syntax nor the morphology nor other relevant factors like the text layout into account. Here the problem is addressed from the contrary perspective. That is, we deal only with features related to the internal text organization represented by syntactic relations as well as by logical document structure of texts. Thus, any lexical information (e.g. like word frequencies) is disregarded. In summary, this paper focuses on the extent to which the structural part alone allows to classify texts.

The text categories investigated in this study represent thematically defined classes or text types which may correspond to genres or registers of texts.

1 For more detail see Power et al. (2003) who argue for treating the abstract document structure as a separate unit of characterization of written texts.
the underlying language, and thus manifest for example official, formal vs. narrative speech. Assuming according to Biber (1995) that linguistic forms vary depending on functional purposes of a register here an attempt is made to guess the category of a text focusing on its formal representation.

In Section 2 the feature selection method, which uses quantitative measures of the text structure representation, is described. In Section 3 we describe the classification method and the evaluation. The two corpora of English and German used for the classification are described in Section 4. In Section 5 the results are presented. In Sections 6 and 7 we summarize the findings and draw the conclusions.

2 Combining structural features with the bag-of-features approach

The term structural feature is related to Köhler (1999) who introduced a number of features applicable to syntactic constituents. In Köhler’s approach, where a syntactic tree representation of texts is assumed, we can calculate among others the following features:2

- **complexity** is defined as the number of immediate daughter constituents of the constituent under consideration (e.g. NP)
- **length** focuses on the subtree rooted by the constituent under consideration and determines the number of its deepest leaves (token)
- **depth** measures the number of production steps from the tree root to the focussed constituent (e.g. from S to NP).

By calculating features as described above for different constituent types on different levels of syntactic embeddedness, he observes regularities concerning for example a sparse complexity of a constituent on one level resulting in a higher complexity on deeper levels. These observations provide an interesting view on language as an open system, like physical systems with interchanging energy.

For the purpose of this paper the features mentioned in the previous passage seem to provide good structural descriptions which according to Biber (1995) should reflect the functional differences of texts in order to discriminate between text types. However, Mehler et al. (2006) who used structural features in a supervised experiment analyzing the 95 rubrics of a German newspaper corpus note that „the relation of structure and function is not deterministic [...]“ making the classification task „anything but trivial“. In their study Mehler et al. (2006) extend the application of the

---

2Köhler (1999) proposes a larger number of features than the three types that are mentioned here.
structural features on syntax to the level of the logical document structure\(^3\) calculating the features from text paragraphs, subdivisions, headings etc. as a sort of constituent-like structure elements. In the study presented here results from the constituent based features as well as those based on the logical document structure are compared.

Given two kinds of text corpora consisting of well defined text categories which are described in Section 4 the features representing a text were extracted in the following way.

1. In the first step for each text constituent type (e.g. a sentence, a paragraph, NP etc.) represented as a tree (see Figure 1.2) the complexity, the depth and the length were computed.

2. In the second step the mean, the standard deviation and the entropy (based on the values calculated in step 1) were obtained for every constituent type and stored in the document vector.

Figure 1.1 shows the feature representation for one constituent, namely \(S\), in a text document. Now the feature vectors for each corpus document and for each structural element can be calculated and applied to classify texts into categories.

Formally, each document vector is represented as a three-dimensional array \(D_{i,j,k}\), where each index \(i, j, k \in \mathbb{N}\) runs over one of the feature sets \(S, T, Q\). \(S\) consists of \(N\) elements and represents an ordered set of structural elements (e.g. sentences, paragraphs etc.). \(T\) is also an ordered set of size \(M\) representing the tree characteristics of Köhler (1999) (like complexity, length, etc.). Finally, \(Q\) has \(L\) elements whereas each of them gives a quantitative measure (mean, standard deviation, etc.). Thus, the feature vector \(\vec{d}\) of a document consists of triples \((i, j, k)\) of \(D\) and has the length \(N \times M \times L\). Each path on Figure 1.1 going from \(S\) to any element of \(Q\)

---

\(^3\)Approaches investigating structural classifiers which make use of the logical document structure are presented e.g. in (Mehler et al., 2005), (Lindemann and Littig, 2006).
represents one component of \( \vec{d} \) corresponding to the triple \( \langle i, j, k \rangle \) and has a real value.

3 Classification Scenario

The classification method used here is based on unsupervised learning\(^4\) which applies Cluster Analysis\(^5\) to divide text documents according to the feature vectors \( \vec{d} \) they are represented by. The main goal of this study is to test the possibilities of a classification using structural features and not to compare different machine learning approaches in this area. A comparison between supervised and unsupervised learning in the structural context is presented in (Pustylnikov and Mehler, 2007).

3.1 Cluster Analysis

*Unsupervised* means that there is no previous knowledge about positive and negative examples of categories as it is the case for *supervised* learning. It also can be described as

"[…] the activity of dividing a set of objects into a smaller number of classes in such a way that objects in the same class are similar to one another and dissimilar to objects in other classes. The classes are not known a priori but have to be discovered;“ Gordon (1987)

In our case we perform 4 clustering experiments, two on each of the corpora.

First, we try out different clustering methods (hierarchical, k-means) with all linkage possibilities (complete, single, average, weighted) and apply various types of distance measures (euclidean, seuclidean, mahalanobis, cityblock, hamming, minkowski, cosine, spearman, jaccard, correlation, chebychev) in order to find the combination achieving the best result. In all kinds of experiments the actual number of categories (i.e. 4 or 7 in our case) in which the documents are to be split is predefined. This is the only kind of information about the categories\(^6\) we provide to the classifier in advance. Normally, one have to decide after every step whether the documents are optimally split and define a break-even point if it is optimal. But knowing the number of categories we define the break-even point in advance in order to reduce the computation. Thus, while the number

\(^4\)See recently performed experiments using unsupervised learning for a structural classification task in the area of hypertext documents: (Mehler et al., 2005).

\(^5\)The idea of clustering texts in terms of their quantitative characteristics is related to (Liiv and Tuldava, 1993). See also (Tambouratzis et al., 2003).

\(^6\)apart from the restriction on category size for the random clustering baseline - see below.
of desired clusters is known by the system a priori, strictly speaking our method should be called *semi-supervised*.

Second, we compare the clustering results against a baseline of random clustering. The baseline is computed by using the same document vectors $\vec{d}$ and by clustering them randomly into the predefined number of categories of the same size as the original categories. Thus, by preserving the original category size we make the clustering not completely random, but rather reflecting the original partition of categories.

### 3.2 Evaluation of the Classifier

The goodness of the classification is measured by the *F-Measure*, the harmonic mean between *precision* and *recall*. Precision and recall are terms commonly used in information retrieval to evaluate the quality of classifications. Precision gives the rate of correctly classified documents to all the documents classified to a category. Recall gives the rate of correctly classified documents to the total number of possible correct documents. The two measures range between $[0, 1]$ and are orthogonal to each other. A classification can be judged as good showing both high precision and high recall values for all categories, that is expressed by the *F-Measure* as a harmonic mean of the two values (Baeza-Yates and Ribeiro-Neto, 1999). In our case the *F-Measure* makes judgements about the overall separability of clusters and is a standard measure to evaluate the results of Cluster Analysis (see Hotho et al., 2005).

### 4 Corpora

The experiments were performed using two text corpora of English and German containing thematically defined categories represented by various text documents.

#### 4.1 The SUSANNE Corpus (Sampson, 1995) of English

The SUSANNE corpus of English or its XML version\(^7\) represents four register categories (see Table 1.1).

\(^7\)The XML based version of SUSANNE is available from [http://ariadne.coli.uni-bielefeld.de/indogram/](http://ariadne.coli.uni-bielefeld.de/indogram/).
There are 64 documents of a comparable size available in the corpus, divided into an equal number of 16 documents for each category. The total number of words constitutes 130,000 words and represents a subset of the Brown Corpus of American English (BNC). The corpus contains a rich syntactic annotation allowing to compute the statistic values described above. In order to reduce the computation effort we focus on 18 global constituent types, for example: NP’s, VP’s, etc. Thus, we unify the fine grained SUSANNE distinction (Figure 1.2) of e.g. Nns and Np to a single constituent type N. This way, we also restrict on less specific and thus presumably more significant constituents allowing to discriminate text types. Fine grained subdivisions in contrast, may be too specific and could bring noise in the classification. To build the feature vectors every constituent type presented in Table 1.2 was analyzed in a way described in Section 2. The notations like O, Q, etc. originate from the corpus annotation and represent levels of constituent structural embedding.
4.2 The German Newspaper Corpus from the *Süddeutsche Zeitung*

The SZ corpus used here consists of 7 thematic categories extracted from the sample of the 10 years of *Süddeutsche Zeitung*, a German newspaper. The corpus have been automatically annotated with the *Text Miner* (Mehler, 2004) system which assigns logical structure information to the raw text and stores the output in an adapted version of the *XCES*-XML (Ide et al., 2000) format. The categories with the total number of 5015 documents are listed in Table 1.4.

In contrast to the SUSANNE corpus, the 7 categories of the SZ are not equal in size, which may have influenced the results of the classification. The annotation of the SZ corpus provides no information about the syntactic relations, it only represents the logical document structure that results in a more scarce feature composition than it is the case for SUSANNE. Thus, here we focus on structures like divisions (DIV), paragraphs (P), etc. up to the the sentence level (Table 1.3). Sentences in contrast to SUSANNE are flat structured not distinguishing the different types of phrases. That means that the *complexity* of a sentence corresponds in most cases to its *length*, which is not the case for SUSANNE where the sentence *complexity* gives the number of underlying phrases, and the *length* gives the number of tokens. The *depth* does not vary between SZ documents for all structural elements remaining almost constant, and is therefore disregarded.

In summary, structural features used to classify SZ are mainly reduced to features of *complexity* and *length*. It was interesting to see to what extent the classification result would be influenced by this factor. Finally, two different

---

*This corpus has recently been used for the supervised experiments by (Mehler et al., 2006), in the following abbreviated with SZ.*

---
classification experiments were run, one classifying genres of SUSANNE, other classifying thematic categories of SZ.

5 Results

The results of 4 clustering experiments are presented in Table 1.5. The second column shows the best F-Measure obtained from the clustering experiment on 4 genres of SUSANNE (0.85938) and of the clustering experiment on 7 thematic categories of SZ (0.84799). The last column represents the best F-Measures of the corresponding baseline clustering experiments on the same corpora.

6 Discussion

First of all both F-Measures of ≈ 0.86 are much higher than the baseline F-Measure values of ≈ 0.38. The F-Measure value of SUSANNE with 0.85938 is slightly higher than the value of SZ (0.84799). There may be different reasons for that. On the one hand, the unequal corpus size and the larger number of categories may have played a role. On the other hand, the difference is so small that the negative factor of the inequality in size seems not to disturb the classifier very much and may indicate for its robustness. It seems to be, that dealing with a larger number of categories also does not influence the results.9

7 Conclusion

In this paper we presented a text-classification approach based on text structure. We started from the assumption that different thematic, text type or register specific categories differ structurally. That means that looking at a structural pattern we must be able to guess the category. Thus, we computed statistical characteristics on structural patterns observed in texts and clustered the texts according to these characteristics. We investigate two corpora of English and German, one representing genres (SUSANNE) and one containing thematic fields of press communication (SZ).

The results obtained in the study show much better F-Measure values than the respective baseline results. The fact that the classification for SZ was so successful is also interesting especially against the background of using a smaller number of structural elements and thus a smaller number of distinguishable features than in the case of SUSANNE. This result extends

9See (Mehler et al., 2006) who achieved good results by classifying 95 categories.
the applicability of the described method not only to syntactically annotated
texts, but to any kind of data having structural information accessible.¹⁰

### 8 Outlook

The results reported here offer an alternative way of text-classification by
focusing on structure. The impact of the approach for different types of
structures must be investigated in more detail. This may include the addi-
tion of different structure types, tree (or graph) characteristics as well as a
larger amount of quantitative parameters. That way, we intend to filter out
the most significant structural indicators by disregarding the useless ones.
Further, a detailed comparison of different classification scenarios is needed
in order to provide a deeper exploration of the possibilities and limits of the
approach.

### Bibliography

Reading, Massachusetts: Addison-Wesley.

Comparison.* Cambridge University Press.

the Royal Statistical Society* 150(2), 119–137.

GLDV.

830.


¹⁰ Whereby the existence of the distinguishable categories within the data is assumed.


Managing Structured Data with Controlled English and Description Logics

Camilo Thorne
KRDB Research Centre
Free University of Bozen-Bolzano
3, Piazza Domenicani, 39100 (Italy)
cthorne@inf.unibz.it

Abstract. Retrieving data from a database is a well-defined and unambiguous task, namely that of evaluating a formal query of a limited expressiveness over an instance of the database schema. This query characterizes exactly and unequivocally the data to be retrieved. The same holds for specifying and storing relational data. But database interfaces are often obscure for non-experts and even more so interfaces supporting ontology reasoning services. We believe that this problem can be solved by advocating the use of controlled languages. That is, by defining a fragment of English, Lite English, that compositionally translates into a description logic, DL-LiteR, well-suited for data-management tasks, with very tight expressivity bounds and for which efficient (LOGSPACE) query answering (QA) algorithms exist.

1 Introduction

The tasks of structuring, modelling, declaring, updating and querying data are all but trivial, let alone intuitively appealing to a casual end-user. The interfaces of relational database management systems (RDBMSs) like dBASE or Oracle are based on query languages such as SQL or Datalog which require skills way beyond those of non-experts. This task becomes even harder when we consider adding a reasoning layer over RDBMSs, by using ontologies and advocating an ontology-driven data access strategy (cf. [4]).

This may be avoided by shifting to natural language interfaces as propounded by, e.g., Androustopoulos in [1]. But query and ontology languages are formal languages (combining declarative and imperative features) that allow no place for ambiguity (cf. [12]), whereas natural language is full of ambiguities. In particular, retrieving data of a relational datasource w.r.t. ontology involves satisfying some crucial requirements or constraints, namely:

(i) A formal query must characterize exactly the data to be retrieved from a database – the set of tuples (or records) that satisfy it (cf. [12]).
The expressive power of query languages must be well-known. The problem of query answering (QA) for relational databases (DBs) falls under \textsc{LOGSPACE} w.r.t. data complexity, i.e., on the number of records of the database (cf. [12]).

When we take into account ontologies, QA becomes the entailment $\mathcal{O}, \mathcal{D} \models \psi$, where $\mathcal{O}$ is an ontology, a logical theory about the domain, $\mathcal{D}$ a DB and $\psi$ a formal query. Hence, the expressive power of the ontology language should also be subject to strict expressivity bounds: QA should stay in \textsc{LOGSPACE}.

To address the problem of managing structured data w.r.t. an ontology with NL a compromise between the expressive power and features of query and ontology languages and the intuitive appeal of NL has to be reached. We believe that this compromise involves the use of controlled languages, partly following the ideas suggested by Sowa in [14]. Controlled languages are fragments of NL tailored to deal with data management tasks, which means that they have been stripped of every ambiguity. Their utterances translate into some logical expression that encodes semantics at the sentence level (cf. [14]) and which can then be postprocessed into, say, SQL.

The remainder of this paper is structured as follows. In section 2 we will argue that controlled languages should be combined with the so-called fragments of language approach of I. Pratt and A. Third, which deal explicitly with the issue of expressive power. Section 3 will be devoted to DL-Lite$R$, the logic we consider the best suited to carry on with data management tasks and that we have taken as the basis of our controlled language. Section 4 will then introduce the controlled language, Lite English, we are working in, together with some expressive power results. Last, but not least, Section 5 will outline our conclusions so far and the ongoing work.

2 Fragments of English and Controlled Languages

Controlled languages (CLs) are fragments of natural language, say, of English, with a limited lexicon and set of grammar rules. Their main aim is to perform data management tasks in NL: specifying, declaring and querying data structured and stored in knowledge bases or databases, like, e.g., Attempto Controlled English (ACE) in [11]. They must be able to express three things: (a) the specification or conceptualization (a.k.a. ontology) of the domain, (b) the data and (c) the user queries (cf. [14]). Constraints on components are used to force utterances to have a unique parse tree and to translate into a unique semantic representation. This has the effect of eliminating ambiguity.

Example 2.1. Let us have a look at how ACE works. In ACE grammar rules have the form:
Camilo Thorne

WhQuestion --> NP[+Q,+NOM] VP/-

The features are to be read as follows: a Wh-question must be rewritten into a focus (+NOM) NP containing a wh-word (+Q) and a VP with no gaps (/-). As semantic representation, ACE uses discourse representation structures (DRSs). For example, this would be the DRS associated to an indefinite NP (cf. [11])

\[
\begin{array}{|c|}
\hline
\text{structure(x, dom)} \\
\text{quantity(x, unspec, unspec, y, eq, unspec)} \\
\text{object(x, unspec, person)} \\
\hline
\end{array}
\]

But then, the issue of the expressive power of these semantic representations and how they are computed becomes critical, if we are to satisfy the requirements (i) – (iii) on data management and access we underlined in the introduction. Why? Because these semantic representations may belong to any logic, some of which may lie beyond the reach of any formal query or ontology language for which QA w.r.t. an ontology is tractable. For instance, DRSs are as expressive as FOL, which would imply QE to be undecidable.

CLs have, therefore, to be complemented with fragments of natural language (FLs), an approach focused in measuring the contribution of each syntactic construct of NL to expressive power and computational complexity. This is possible because following formal semantics, their utterances compositionally translate modulo a compositional translation \(\phi\) into a semantic representation called meaning representation (MR). This MR is, typically, a FOL formula. This approach stems from Ian Pratt and Alan Third’s work on English FLs (cf. [13]). Pratt and Third build incrementally a family of English fragments by starting from a fragment, COP, that covers very basic constructs, like copula, nouns, negation and quantification. Its coverage is then extended to other NL constructs:

<table>
<thead>
<tr>
<th>Fragment</th>
<th>Coverage</th>
<th>Decision class for SAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>COP</td>
<td>Copula, common and proper nouns, negation, universal and existential quantifiers.</td>
<td>P</td>
</tr>
<tr>
<td>COP+TV+DTV</td>
<td>COP + transitive verbs, distransitive verbs.</td>
<td>P</td>
</tr>
<tr>
<td>COP+REL</td>
<td>COP + relative pronouns.</td>
<td>NP-Complete</td>
</tr>
<tr>
<td>COP+REL+TV</td>
<td>COP + transitive verbs, relative pronouns.</td>
<td>EXPTIME-Complete</td>
</tr>
<tr>
<td>COP+REL+TV+DTV</td>
<td>COP+TV+DTV + relative pronouns.</td>
<td>EXPTIME-Complete</td>
</tr>
<tr>
<td>COP+REL+TV+RA</td>
<td>COP+REL+TV + restricted (intrasentential) anaphora.</td>
<td>NEXPTIME-Complete</td>
</tr>
<tr>
<td>COP+REL+TV+GA</td>
<td>COP+REL+TV + generalized anaphora.</td>
<td>undecidable</td>
</tr>
</tbody>
</table>
The expressive power of a FL is then defined as that of the fragment of FOL into which it compositionally translates. This is because the set of MRs of each fragment constitutes a fragment of FOL. The key idea is that each NL construct adds a new logic construct to the underlying FOL fragment, modifying both its expressive power and its computational properties. In particular, each addition affects the complexity of the satisfiability (SAT) and entailment problems of the logic fragments, until they become undecidable. Note further that as they are closed under boolean negation, entailment reduces to SAT.

Example 2.2. Some examples may help at this point. As the reader can see, each utterance gives way to a FOL meaning representation exhibiting different logical constructs, following the semantics of function words and content words in each fragment (cf. [13]):

<table>
<thead>
<tr>
<th>Sentence</th>
<th>MR (FOL)</th>
<th>Fragment</th>
</tr>
</thead>
<tbody>
<tr>
<td>No man is a woman</td>
<td>∀x(\text{Man}(x) \rightarrow \neg \text{Woman}(x))</td>
<td>COP</td>
</tr>
<tr>
<td>Every man who is not dead is alive</td>
<td>∀x(\text{Man}(x) \land \neg \text{Dead}(x) \rightarrow \text{Alive}(x))</td>
<td>COP+</td>
</tr>
<tr>
<td>Every scholar reads some book</td>
<td>∀x(\text{Scholar}(x) \rightarrow \exists y(\text{Book}(y) \land \text{Reads}(x, y)))</td>
<td>COP+REL</td>
</tr>
<tr>
<td>Every salesman sells some merchandise to some customer</td>
<td>∀x(\text{Salesman}(x) \rightarrow \exists y(\text{Customer}(y) \land \exists z(\text{Merchandise}(z) \land \text{Sells}(x, y, z))))</td>
<td>COP+TV+DTV</td>
</tr>
</tbody>
</table>

The complexity of analysis I. Pratt and A. Third above show that only the first two fragments, COP and COP+TV+DTV are tractable. But, why is this the case? Because of the logic constructs these NL constructs can express or capture. A careful glance at the fragments tells us the following:

- Quantifiers may occur in any order.
- Negation expresses boolean negation.
- Relative pronouns express boolean conjunction.
- Transitive verbs express binary relations. Ditransitive verbs, ternary relations. Nouns, unary relations (i.e., sets).

The introduction of relative clauses produces an exponential blowup, and therefore intractability. Why? Because COP+REL can express boolean conjunction and negation. That is, a complete set of boolean operators, whence MRs become as expressive as the propositional calculus which is NP-Complete. On the other hand, introducing binary and ternary relations does not affect computational complexity.

Recall that it is essential (requirement (iii)) for entailment to be tractable. This means that, if we want to retain this property, we have to stick, as far as possible, to the English constructs that COP and COP+TV+DTV cover. As we will see, this is actually the case when we define a FL or a CL that compositionally translates into DL-LiteB.
3 DL-Lite and QA

DL-Lite\textsubscript{R} (cf. \cite{7}) is a description logic specifically tailored to meet data management requirements and tasks and in particular, to query relational data sources w.r.t. an ontology (cf. \cite{5}). DL-Lite\textsubscript{R} allows us to encode ontologies and databases as logical theories called knowledge bases which can then be queried with simple formal queries called conjunctive queries. Querying is defined in terms of logical entailment. Given that both conjunctive queries and DL-Lite\textsubscript{R} are decidable fragments of FOL, this definition makes sense. Moreover, QA can be decided efficiently. In this way, it satisfies the requirements (i) – (iii). It is, as a matter of fact, a maximal description logic for which this property holds.

Definition 3.1. (Concepts) Let \( P = \{ P_i | i \in \mathbb{N} \} \) and \( R = \{ R_i | i \in \mathbb{N} \} \) be two countable sets of primitive concept and role symbols. DL-Lite left hand side concepts \( C_l \) and right hand side concepts \( C_r \) are defined as follows:

\[
C_l := P \mid \exists R \mid \exists R^{-} \mid C_l \cap C_l.
\]

\[
C_r := \neg P \mid \neg \exists R \mid \neg \exists R^{-} \mid C_l \mid C_r \cap C_r \mid \exists R : C_r \mid \exists R^{-} : C_r.
\]

Concepts stand for sets. Role symbols for binary relations. \( R^{-} \) stands for the inverse of \( R \). Concepts of the form \( \exists R \) are known as unqualified existential roles, and are built by existentially quantifying the second argument of the relation. Concepts of the form \( \exists R : C \) are called qualified existential roles and are similar to unqualified roles, only that this time we assert as well that the quantified argument falls under concept \( C \).

Definition 3.2. (Assertions) Let \( C_o = \{ c_i | i \in \mathbb{N} \} \) be a set of constants. DL-Lite facts \( A \) and terminological assertions \( T \) are defined as follows:

\[
A := P(C_o) \mid R(C_o, C_o) \quad \text{(facts)}
\]

\[
T := C_l \sqsubseteq C_r \mid R \sqsubseteq R \quad \text{(terminological assertions)}
\]

Concept subsumption \((\sqsubseteq)\) stands for set inclusion. Assertions are bundled into knowledge bases (KBs): tuples of the form \( \langle \text{ABox}, \text{TBox} \rangle \), where the ABox is a set of facts and the TBox a set of terminological assertions. The TBox encodes, intuitively, the ontologies or conceptual models of the data (the data constraints) and the ABox the actual data to be declared or stored. The size of an ABox is given by the number of pairwise distinct constants of its ABox, denoted \#(ABox). This notion is a.k.a. the data complexity of a KB.

Example 3.1. The following is an example of a DL-Lite\textsubscript{R} KB. We want to specify in a KB \( K_0 \) some properties of the domain of men, limited to a single individual, James. These properties are their being mortal and their owning a car:
We can extract data from KBs by using formal queries. They are queried with conjunctive queries:

**Definition 3.3. (Conjunctive Queries)** A conjunctive query (CQ) is an expression of the form \( q(\vec{x}) \leftarrow \exists \vec{y} \beta(\vec{x}, \vec{y}) \) where \( \vec{x} \) is a possibly empty finite sequence of distinguished variables and \( \beta(\vec{x}, \vec{y}) \) (the body) a conjunction of atoms over \( \vec{y}, \vec{x} \) using DL-Lite basic roles and concepts, as well as individual constants. Relation \( q \) is its head. If \( \vec{x} \) is empty, the query is said to be boolean.

As a matter of fact, a CQ is basically a FOL open formula with \( k \) free variables, where \( k \) is the arity of its head. They are also equivalent to the SELECT-PROJECT-JOIN fragment of SQL. We can now formally define QA for DL-LiteR: it consists in computing the answer set of a CQ \( q \) over an ABox w.r.t. a TBox and, ultimately, in deciding the entailment stated in (iii):

**Definition 3.4. (Query Answering)** The semantics of a CQ \( q \) is the set of constant sequences \( \vec{c} \) such that the logical entailment \( \langle \text{TBox}, \text{ABox} \rangle \models q(\vec{c}) \) holds – where \( q(\vec{c}) \) denotes the grounding of the CQ \( q \) w.r.t. \( \vec{c} \).

**Example 3.2.** Consider the following question: "Who owns a car?". The corresponding CQ is: \( q := q(x) \leftarrow \exists y \text{Owns}(x, y) \). If we evaluate this query over \( K_0 \) we will get as answer set \( \{ \text{James} \} \), since \( K_0 \models q() \leftarrow \exists y \text{Owns}(\text{James}, y) \). Note that the grounding of \( q \) is the boolean query obtained from \( q \) by erasing its distinguished variable and applying the closed substitution \([\text{John}/x]\) on its body.

But this entailment problem would not be interesting for our purposes if it had not the right properties. The fundamental result is that it does. It can be decided quite efficiently (w.r.t. data complexity):

**Theorem 3.1. (Calvanese et. al. [5])** Deciding whether \( \langle \text{TBox}, \text{ABox} \rangle \models q(\vec{c}) \) holds is LOGSPACE on data complexity that is, on \( \#(\text{ABox}) \).

Moreover, DL-LiteR is a tractable logic: TBox satisfiability is in P, which means that it is similar in expressivity to the tractable fragments of English. It is actually a fragment of FOL (cf. [2]) with some syntactic restrictions. Restrictions that explain why QA and SAT can be decided so efficiently, as opposed to other, more expressive, description logics (cf. [2, 10, 5]): terminological assertions are assumed to be (implicitly) universally quantified. There are no variables. But, most importantly: negation occurs only to the right of \( \sqsubseteq \). Furthermore, facts contain no negation. Last, but not least, DL-LiteR assertions belong to the \( \forall \exists^* \) FOL prefix class: quantifier prefixes occur in a fixed order.
4 Lite English

The second step to tackle the problem of managing data with NL involves defining a controlled language, that we have called Lite English (cf. [3]), and studying its expressivity. Why? Because now that we have chosen a suitable logic, we need to see which NL constructs can be safely translated into DL-LiteR concepts and assertions. Moreover, since SAT for any DL-LiteR TBox is in P, we must make clear to what extent Lite English differs from COP and COP+TV+DTV – the tractable FLs. This latter issue is particularly crucial, since it helps us in identifying the NL constructs that play a distinctive role in QA. Lite English has been “reverse-engineered” from DL-Lite and consists in two fragments:

- The declarative fragment of declarative sentences, based on DL-LiteR. It translates into TBox and ABox assertions.
- The interrogative fragment of Wh and Y/N-questions based on the CQs. Wh-questions are translated into CQs and Y/N-questions into boolean CQs with no free variables.

Clearly, as we use a compositional translation in the spirit of Pratt and Third, this implies that the expressive power of these fragments (and of Lite English itself) is simply that of CQs and DL-Lite. Clearly, Lite English satisfies data management constraints (i) – (iii) and captures QA in a fragment of English.

Example 4.1. The following table gives an idea of what we mean by the utterances Lite English can capture:

<table>
<thead>
<tr>
<th>Lite English</th>
<th>Fragment</th>
<th>MR (DL-Lite+CQs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Every salesman sells some merchandise that is</td>
<td>declarative</td>
<td>(\text{Salesman} \sqsubseteq \exists \text{Sells}: \text{Merchandise} \cap \neg \text{Expensive} )</td>
</tr>
<tr>
<td>John is uninteresting</td>
<td>declarative</td>
<td>(\text{Uninteresting}(\text{John}) )</td>
</tr>
<tr>
<td>No man is a woman</td>
<td>declarative</td>
<td>(\text{Man} \sqsubseteq \neg \text{Woman} )</td>
</tr>
<tr>
<td>Everybody who likes something succeeds</td>
<td>declarative</td>
<td>(\exists \text{Likes} \sqsubseteq \text{Succed} )</td>
</tr>
<tr>
<td>Who knows Roger?</td>
<td>interrogative</td>
<td>(q(x) \leftarrow \text{Knows}(x, \text{Roger}) )</td>
</tr>
<tr>
<td>Does Julian rule?</td>
<td>interrogative</td>
<td>(q() \leftarrow \text{Rules}(\text{Julian}) )</td>
</tr>
</tbody>
</table>

4.1 Constraining Parse Trees

The compositional translation \(\phi\) works as follows. First, higher order logic (HOL) expressions are associated to the words both of the function and of the content lexicon. Then, we mirror the composition of syntactic components in the parse tree with lambda application and reduction, ultimately yielding a FOL meaning representation at the sentence level, following the pattern set by Clifford in [9]:
In this parse tree of the Y/N-question "Is James a Man?", the feature structure \( \text{sem} \) returns the current values of the compositional translation \( \phi \) at each node of the tree: The MR (the feature \( \text{mr} \)), a CQ of type \( t \), is computed w.r.t. a semantic type (the feature \( \text{type} \)) and a context (the feature \( \text{con} \)), following the usual HOL assumptions, advocated since Montague, regarding the type associated to every NL component – like \( (e \to t) \to t \), i.e., a function from properties onto truth values, for NPs or \( e \to t \), a property or characteristic function, for Ns. The fact that the root context is empty implies that the whole expression (i.e., the MR) is well-typed (cf. [8]), ensuring the termination of the translation procedure.

The feature structure \( \text{syn} \), on the other hand, returns the category (\( \text{cat} \)), the position (\( \text{pos} \), which can be bound to two values, \( \text{subj} \), subject, and \( \text{pred} \), predicate) and the kind of utterance (\( \text{ass} \), bound to \( \text{abox} \), \( \text{tbox} \) and \( \text{cq} \), i.e., to ABox and TBox assertions and CQs) of every non terminal component and of every function word. The \( \text{gap} \) feature associated to NPs indicates whether it contains or not the trace of a wh-movement (in the case of subordinate clauses) and the \( \text{coord} \), if it contains a conjunction. A voice (for voice) feature, that can be set to \( \text{act} \) (active voice) or \( \text{pass} \) (active voice), is associated to VPs.

The grammar we have been using so far is a unification-based phrase structure grammar (UPSG), where nested feature structures comprising both syntactic and semantic features are associated to (and computed w.r.t.) each
component. They have the advantage that parsing is based on constraint satisfaction and that it is thus easy to set constraints on components by means of semantic and syntactic features.

4.2 Expressive Power of the Declarative Fragment

We have studied the expressivity of the declarative fragment by comparing DL-LiteR’s expressive power to that of the sets of MRs of the fragments COP and COP+DTV+TV, which we will denote, respectively, \( \Lambda_{\text{COP}} \) and \( \Lambda_{\text{COP+DTV+TV}} \). This is based on the following standard model-theoretic characterization of the expressive power of a FOL fragment:

**Definition 4.1. (Expressive Power)** The expressive power of a fragment \( \Lambda \) of FOL over a signature \( L \) is defined in terms of the class \( K_\Lambda \) of the FOL models or interpretation structures over \( L \).

A FOL fragment \( \Lambda' \) is said to be as expressive as or to contain a fragment \( \Lambda \) if \( K_\Lambda \subseteq K_{\Lambda'} \) – i.e., if it can express all of the models of \( \Lambda \). If \( K_\Lambda \cap K_{\Lambda'} \neq \emptyset \), they are said to overlap. Armed with these definitions, we can state the main results regarding Lite English’s expressive power:

**Theorem 4.1.** Lite English is as expressive as COP if we drop DL-Lite’s unique name assumption, but the converse is false.

**Theorem 4.2.** Lite English and COP+TV+DTV overlap w.r.t. expressive power but neither is as expressive as the other.

We will not give the proofs here, which the reader can find in [2, 15]. The general picture can be summarized as follows:

![Diagram showing expressive power overlap]

The most interesting consequence of these results is the light they shed on the NL constructs Lite English covers, in particular, the use of relatives in the declarative fragment. We can summarize the relevant features of Lite English declarative sentences as follows:
• (Quantification) In DL-Lite$_R$ and a fortiori in Lite English, a universal quantifier can be followed by $n \geq 0$ existential quantifiers. This is not the case in COP+TV+DTV. In this fragment, quantifiers may occur in any order whatsoever. Statements like "some woman loves every man" are in COP+TV+DTV but not in Lite English.

• (Negation) As the reader may recall from the definition of ABox assertions, DL-Lite$_R$ precludes negated facts (i.e. negated ABox assertions). Negation in DL-Lite can moreover only occur on right hand side TBox concepts. But in COP and a fortiori in the fragment COP+TV+DTV this is possible - statements like "Julian is not an emperor" are in COP but not in Lite English. Negation is thus highly controlled.

• (Relatives) English Lite covers constructs that neither the fragment COP nor COP+TV+DTV can cover without yielding a state blowup. Lite English supports, for instance, relative clauses, without compromising tractability, as it happens with FLs (recall Pratt and Third’s table above). Lite English outrules an unrestricted (i.e., uncontrolled) use of negation, which can only occur within the predicate of a general statement, mirroring DL-Lite$_R$.

• (Relations) Lite English covers only transitive verbs (binary relations), even if, in principle, nothing prevents us from extending coverage to ditransitive verbs: the properties of DL-Lite$_R$ hold for $n$-ary relations (cf. [6]). However, due to the restricted behavior of quantifiers a good many properties of relations that can be expressed by, say, COP+TV+DTV, will not be expressible by DL-Lite$_R$.

5 Conclusions and Further Work

We have studied the problem of managing structured data with natural language and propounded a controlled language (Lite English) approach to it based on description logics, DL-Lite$_R$. We have shown how this language, or at any rate its declarative fragment, inherits the expressivity of DL-Lite$_R$. It is thus, in theory, suitable for carrying on with data management tasks respecting their tight expressivity bounds. We have furthermore compared the expressive power of Lite English with that of the tractable fragments of English, singling out the NL constructs relevant to ontology-driven data access.

The main issue on which we are working now is that of applying the same techniques to the interrogative fragment, whose expressive power corresponds to that of conjunctive queries. The contribution of the NL constructs we encounter in questions to expressive power and computational complexity is still, as far as we know, an open question.
Bibliography


Some clarifications in logics of agency

NICOLAS TROQUARD
Institut de Recherche en Informatique (Toulouse, France)
Laboratory of Applied Ontology (Trento, Italy)
troquard@irit.fr

ABSTRACT. We review the logic of “seeing to it that” (STIT). We propose two new primitive operators that allow to characterize syntactically the operators Chellas’ deliberative and achievement stit but also Chellas’ original operator of agency $\Delta_a\varphi$. We show how it highlights their relationship and reveal differences. In particular, we remark that Chellas’ stit is not an accurate simulation of Chellas’ $\Delta_a\varphi$.

1 Introduction

Recently, the STIT theory has gained interest in the field of logics for computer science and artificial intelligence [Wan06, BHT06] and in ontology [TTV06, Gar06]. It is worth noting that it will be central in the introductory course “Logics of Agency and Multi-Agent systems” of ESSLLI 2007.

STIT originates from philosophy. Probably the first paper to refer to the logic of seeing to it that (or theory of agents and choices in branching time) is [BP88]. It analyzes linguistically the needs for a general theory of “an agent making a choice among alternatives that lead to an action”. The thesis is that the best way to meet this goal is to augment the language with a class of sentences. The proposed class is the one of sentences of the form “Ishmael sees to it that Ishmael sails on board the Pequod” paraphrasing the sentence “Ishmael sails on board the Pequod”. Thus, from any sentence describing a concrete action of an agent $a$ (e.g., sailing) we can reformulate it into an agentive one stating that $a$ sees to it that a state of affairs $\varphi$ holds, formally: $[a \text{ stit}: \varphi]$.

Formal models are provided, that constrain those of the oldest semantics for a logic of action introduced by Chellas in [Che69], such that time is linear to the past. Several agents with independent choices are also assumed. However, Belnap et al. release the assumption of discreteness. It is important to remark that models are influenced by the observation that in a branching time framework, future-tensed statements are ambiguous to evaluate if not impossible. In general, in branching time, a moment...
Some clarifications in logics of agency

alone does not provide enough information to determine the truth value of a sentence about the future. Prior [Pri67] and Thomason [Tho70] hence proposed to evaluate future-tensed sentences with respect to a moment and a particular course of time running through it. This is why states of the world in STIT models consist of ‘fragmentized’ moments: a moment splits up into as much indexes as there are courses of time running through it.

[BP88] is a roadmap towards a very rich theory of agency compiled in [BPX01] and [Hor01]. One of the core ideas is to capture a notion of responsibility of the agent $a$ for the actual truth of a proposition $p$.

In this note, we review STIT theory (Section 2) and propose two new primitive operators that allow to characterize syntactically the operators Chellas’, deliberative and achievement stit (Section 4) but also Chellas’ original operator of agency $\Delta_a \varphi$. We show how it highlights their relationship and reveal differences. In particular, we remark in Section 5 that Chellas’ stit is not the more accurate simulation of Chellas’ original proposal $\Delta_a \varphi$ [Che69, Che92]. A brief preliminary investigation of duration of agents’ activities is given in Section 6, and we conclude in Section 7.

2 The theory of agents and choices in branching time

We present here the semantics provided by Horty and Belnap [HB95].

It is embedded in the branching time framework. It is based on structures of the form $\langle W, < \rangle$, in which $W$ is a nonempty set of moments, and $<$ is a tree-like ordering of these moments. A maximal set of linearly ordered moments from $W$ is a history. Thus, $w \in h$ denotes that the moment $w$ is on the history $h$. We define $\text{Hist}$ as the set of all histories of a STIT structure. $H_w = \{h | h \in \text{Hist}, w \in h\}$ denotes the set of histories passing through $w$. An index is a pair $w/h$, consisting of a moment $w$ and a history $h$ from $H_w$ (i.e., a history and a moment in that history). Because of branching, two different moments can lie at a same instant. In the following $\text{Agt}$ is a non-empty set of agents and $\text{Atm}$ is a set of atomic propositions. A STIT-model is a tuple $\mathcal{M} = \langle W, <, \text{Choice}, \text{Instant}, v \rangle$, where:

- $\langle W, < \rangle$ is a branching time structure;
- $\text{Choice} : \text{Agt} \times W \rightarrow 2^{2^{\text{Hist}}}$ is a function mapping each agent and each moment $w$ into a partition of $H_w$. The equivalence classes belonging to $\text{Choice}_a^w$ can be thought of as possible choices or actions available to $a$ at $w$. Given a history $h \in H_w$, $\text{Choice}_a^w(h)$ represents the particular

\footnote{For any $w_1$, $w_2$ and $w_3$ in $W$, if $w_1 < w_3$ and $w_2 < w_3$, then either $w_1 = w_2$ or $w_1 < w_2$ or $w_2 < w_1$.}
choice from $\text{Choice}_a^w$ containing $h$, or in other words, the particular action performed by $a$ at the index $w/h$. We must have $\text{Choice}_a^w \neq \emptyset$ and $Q \neq \emptyset$ for every $Q \in \text{Choice}_a^w$:

- **Instant**: $\text{Instant} : W \rightarrow 2^W$ : maps each moment to the set of moments lying in the same instant. It may be seen as a partition “by layers” of $W$ into equivalence classes;
- $v$ is valuation function $v : \text{Atm} \rightarrow 2^{W \times \text{Hist}}$.

Those models are originally called $BT + I + AC$ structures, explicitly listing their main characteristics, viz. branching time, instants, agents and choices.

In STIT-models, moments may have several valuations, depending on the histories passing through them. Thus, at any specific moment, we have different valuations corresponding to the results of the different (non-deterministic) actions possibly taken at that moment.

A formula is evaluated with respect to a model and an index. Here are basic truth conditions:

- $\mathcal{M}, w/h \models p \iff w/h \in v(p), p \in \text{Atm}.$
- $\mathcal{M}, w/h \models \neg \varphi \iff \mathcal{M}, w/h \not\models \varphi.$
- $\mathcal{M}, w/h \models \varphi \land \psi \iff \mathcal{M}, w/h \models \varphi$ or $\mathcal{M}, w/h \models \psi.$

Historical necessity (or inevitability) at a moment $w$ in a history is defined as truth in all histories passing through $w$:

- $\mathcal{M}, w/h \models \Box \varphi \iff \mathcal{M}, w/h' \models \varphi, \forall h' \in H_w.$

There are several operators in the STIT theory. The so-called *achievement stit* was first introduced. Then Hory simplified the logic by introducing a *deliberative* one, which is deprived of the temporal aspect featured by instants, and corresponds to the previous proposition of von Kutschera [vK86, HB95]. We also present the widely used and simpler Chellas’ stit:

**Definition 1** (Choice equivalence). *Two moments $w_1$ and $w_2$ are $\text{Choice}_a^w$-equivalent if (1) $\text{instant}(w_1) = \text{instant}(w_2)$ (2) $w$ is a moment prior to both $w_1$ and $w_2$ (called witness moment) (3) $w_1$ and $w_2$ lie on histories belonging to the same $\text{Choice}_a^w$ partition.*

- $\mathcal{M}, w/h \models \hspace{1em} [\text{a stit}: \varphi] \iff$ there is a moment $w_1 < w$ such that (for all moment $w_2$, $\text{Choice}_a^{w_1}$-equivalent to $w$, $\mathcal{M}, w_2/h' \models \varphi$ for all $h' \in H_{w_2}$) and (there is some moment $w_3 \in \text{instant}(w)$ such that $w < w_3$ and $\mathcal{M}, w_3/h'' \not\models \varphi$ for some $h'' \in H_{w_3}$)

$[\text{a stit}: \varphi]$ means that agent $a$ has ensured that $\varphi$ holds now by making a choice previously, and if he had made a different choice, $\varphi$ could have been false at the present instant.
Some clarifications in logics of agency

\[ M, w/h \models [\text{adstit} : \varphi] \iff \forall h' \in \text{Choice}_a(h), M, w/h' \models \varphi \text{ and } \exists h'' \in H_\varphi, M, w/h'' \models \varphi \]

\[ M, w/h \models [\text{cstit} : \varphi] \iff \forall h' \in \text{Choice}_a(h), M, w/h' \models \varphi \]

Intuitively \([\text{cstit} : \varphi]\) means that agent \(a\)'s current choice ensures \(\varphi\), whatever the other agents do. \([\text{adstit} : \varphi]\) adds the fact that \(\varphi\) was not settled, so, in a sense, that agent \(a\) is responsible for \(\varphi\). Truth conditions of those operators do not depend on instants. They can be evaluated in simpler models called \(BT + AC\) structures.

3 A discrete time framework

What can be now of interest, is to understand the underlying link between the three main versions of \(STIT\) operator, viz. Chellas’ stit, deliberative stit and achievement stit. The deliberative stit can be defined from Chellas’ plus historical necessity since the following holds:

\[ [\text{adstit} : \varphi] \leftrightarrow [\text{cstit} : \varphi] \land \neg \Box \varphi \]

The other way round, we have \([\text{cstit} : \varphi] \leftrightarrow [\text{adstit} : \varphi] \lor \Box \varphi\). The link between deliberative and Chellas’ stit is then quite obvious. However, a formal link of the achievement stit with them is more involved. We nevertheless claim that, because of the complex semantics of \([\text{astit} : \varphi]\), such a relationship can provide a neat picture of the fundamental aspects of the theory of choice in time. And in order to stick to the fundamental aspects, let us first simplify the framework by some usual assumptions. They at least are usual in a discipline like computer science, and have the merit to rule out some features that were enabled just for a matter of generality, and thus unfortunately hid some other essential features. Belnap and colleagues refrained from taking position on the nature of time.

“[...] no assumption whatsoever is made about the order type that all histories share with each other and with \(Instant\). For this reason the present theory of agency is immediately applicable regardless of whether we picture succession as discrete, dense, continuous, well-ordered, some mixture of these, or whatever; and regardless of whether histories are finite or infinite in one direction or the other.” ([BPX01, p. 196].)

We thus consider the assumption of time isomorphic to the set of natural numbers interesting to study. We would like to investigate how such a simplification can strengthen our understanding of logics of agency. We explicit discreteness as follows:

**Definition 2.** The total function \(\text{instantof} : W \rightarrow \mathbb{N}\) associates an instant to each moment. The function \(\text{atinstant} : \mathbb{N} \rightarrow 2^W\) associates each instant to the set of moments lying in.
Hypothesis 1. Histories are regular: (1) $\forall h, h' \in \text{Hist}, \forall w \in h, \exists w' \in h', \text{ s.t. } \text{instantof}(w) = \text{instantof}(w')$ (2) for some $h \in \text{Hist}$ and $w \in h$, if $\text{instantof}(w) = i$ then $\forall j < i, \exists w' \in h \text{ s.t. } \text{instantof}(w') = j$.

Moreover, we assume the existence of a root:

Hypothesis 2. There is a moment $w$ such that there is no $w'$ such that $w' < w$.

Figure 1.1: (Time goes upward.) At $w_0$, $a$ can make the choice that $\varphi$ is true in two steps, even though it is not settled it will be true at that instant. At $w_1$ (or $w_2$) it will be the case $[\text{a stit} : \varphi]$. Indeed, for some $h \in w_1$, $w_1/h \models [\text{a stit} : \varphi]_2$ ($\varphi$ is true at every index of $w_1$ and $w_2$) and $w_1/h \models \neg \Box_2 \varphi$. At $w_0$ it is however already settled that in three steps, $\varphi$ will be false: for some $h' \in w_3$, $w_3/h' \models \Box_3 \neg \varphi$. ($\varphi$ is true at every (upper) dark grey moment.)

4 \(\text{\textit{NSTIT}}\)

In order not to get confused let us call \(\text{\textit{NSTIT}}\) the logic interpreted by $BT + I + AC$ structures constrained by the hypothesis previously presented, and syntactically extending the STIT theory presented in Section 2 (with a language containing operators Chellas’), deliberative and achievement stit) with the two following collections of operators indexed by a natural number $k$:

- $\mathcal{M}, w/h \models \Box_k \varphi \iff \exists w_0 \leq w, \text{instantof}(w_0) = \text{instantof}(w) - k, \forall w' \in \text{Instant}(w) \cap (\bigcup_{h' \in H_{w_0}} h'), \forall h' \in w', \mathcal{M}, w'/h' \models \varphi$

It reads that “$k$ instants ago, it was settled that $\varphi$ would be true now”. 247
Some clarifications in logics of agency

\[ \mathcal{M}, w/h \models [a \text{cstit}: \varphi]_k \iff \exists w_0 \leq w, \ \text{instantof}(w_0) = \text{instantof}(w) - k, \ \forall w' \text{ Choice}^{\text{sto}}_a - \text{equivalent of } w, \ \forall h' \in w', \ \mathcal{M}, w'/h' \models \varphi \]

It reads that “\( k \) instants ago, agent \( a \) ensured that \( \varphi \) would be true now”.

Analogously to the achievement stit, we call \( w_0 \) in the previous truth conditions the witness moment of \([a \text{cstit}: \varphi]_k\) or \( \square_k \varphi \).

We offer to NSTIT a mechanism close to what exists in Hybrid Logic [BdRV01, Chap. 7]. We assume the existence of a set \( \{0, 1, \ldots\} \) of specific atomic formulae that we could call nominals. We thus constrain the models such that \( \mathcal{M}, w/h \models i \iff \text{instant}(w) = i \). Our account is nevertheless different from Hybrid Logic since genuine nominals should be true at exactly one moment/history pair. (See for example [BG01] for a concrete account of hybrid temporal logic.)

Now, let us exhibit some interesting validities, candidates to the status of axioms for future developments.

\[
\begin{align*}
\text{(NP)} & \quad 0 \rightarrow \neg \square_1 \top \\
\text{(P)} & \quad 0 \lor \square_1 \top \\
\text{(Mon)} & \quad \square_{k+1} \top \rightarrow \square_k \top \\
\text{(Sett-1)} & \quad \square_k k \rightarrow \square_k \top \\
\text{(Sett-2)} & \quad k \leftrightarrow \square_k k
\end{align*}
\]

\( \text{(NP)} \) captures that there is no past beyond the instant 0. \( \text{(P)} \) on the contrary means that whenever we do not stand at instant 0 we can ‘step back’ in the temporal structure. \( \text{(Mon)} \) means says that when we can look back at \( k+1 \) steps, we can look back at \( k \) steps as well. \( \text{(Sett-1)} \) says that \( k \) times ago, it was settled that we would be standing at instant \( k \) only if we can look back at \( k \) steps. \( \text{(Sett-2)} \) means that we are standing at instant \( k \) iff it was already settled \( k \) steps ago that we would stand at instant \( k \) now.

We are now ready to see how the operators of the STIT language relate to our new primitives.

**Proposition 1.** The four following formulae are valid:

- \( \square \varphi \leftrightarrow \square_0 \varphi \)
- \( [a \text{cstit}: \varphi] \leftrightarrow [a \text{cstit}: \varphi]_0 \)
- \( [a \text{dstit}: \varphi] \leftrightarrow [a \text{cstit}: \varphi]_0 \land \neg \square_0 \varphi \)
- \( i \rightarrow ([a \text{astit}: \varphi] \leftrightarrow \bigvee_{k=1}^i ([a \text{cstit}: \varphi]_k \land \neg \square_k \varphi)) \)
From the last item, we can have a local definition of achievement stit for every instant. It is indeed similar to the definition of tense operator ‘until’ and ‘since’. (See [BG01, Sect. 4.1].) Historical necessity, Chellas’ stit and deliberative stit on the other hand, can be completely defined from our new primitives.

Instances of the new primitive operator of agency are intrinsically related and obey the following property:

**Proposition 2.** \([acstit: \varphi]_{k_1} \rightarrow [acstit: \varphi]_{k_2}, \text{ for every } k_2 < k_1.\)

5 Comments on Chellas’ \(\Delta_a \varphi\)

In [Che92], Brian Chellas turns back to his operator of agency introduced in [Che69]. As in theories of agents and choices in branching time, truth values of the language are in terms of times (alias instants), histories and agents, plus certain relations. Here, we quickly show how we can define \(\Delta_a \varphi\) fairly in \(NSTIT\), and also suggest that Chellas’ stit operator does not match perfectly.

5.1 Semantics of time and actional alternatives

The set of times is taken to be the set of integers. We write \(t < t'\) to state that \(t\) is earlier than \(t'\) and \(t \leq t'\) to state it is not later. Histories are functions from times to states of affairs (alias moments), and \(h(t)\) represents the state of affairs in history \(h\) at time \(t\). Two time-indexed relations between histories are then defined. \(h =_t h'\) means that histories \(h\) and \(h'\) have the same past at time \(t\); \(h \equiv_t h'\) means that they share the same past and the same present. Formally,

- \(h =_t h'\) iff \(h(t') = h'(t')\) at every time \(t' < t\)
- \(h \equiv_t h'\) iff \(h(t') = h'(t')\) at every time \(t' \leq t\)

Given a state of affairs \(h_t\), Chellas uses the term *future cone* for the set of histories emanating from \(h_t\). Two histories are in the future cone of \(h(t)\) if \(h \equiv_t h'\).

**Instigative alternatives.** The relation \(R^a_t(h, h')\) is used to mean that \(h'\) is an *instigative alternative* of \(h\) for agent \(a\) at time \(t\). The relation is reflexive and \(R^a_t(h, h')\) only if \(h =_t h'\). Instigative alternatives capture the idea of histories “under the control” or “responsive to the action” of \(a\) at \(t\).

Truth conditions of the operator of agency is given by:

\[
(h, t) \models \Delta_a \varphi \iff (h', t) \models \varphi, \forall h' \text{ s.t } R^a_t(h, h')
\]
5.2 Chellas’ stit is not $\Delta_a \varphi$

In addition to our short overview, it is interesting and helpful to consider Krister Segerberg’s interpretation of the operator in [Seg92]. Segerberg calls $R^a_t(h, h')$ the cone of ‘actional alternatives’ and observes that in the truth value of $\Delta_a \varphi$, “the cone Chellas wishes to consider has its apex at the immediately preceding time”. This is indeed a consequence of the constraint that two histories $h$ and $h'$ are instigative alternatives only if $h \equiv_t h'$.

Finally, we can define more appropriately the operator in NSTIT as follows:

$$\Delta_a \varphi \equiv [a \text{cstit}: \varphi]_1$$

It thus clearly differs from $[a \text{cstit}: \varphi]$ which we remind is logically equivalent in NSTIT to $[a \text{cstit}: \varphi]_0$. There is a temporal switch between them. One must be aware of a possible misconception of the Chellas’ stit, since it does not reflect Chellas’ original operator. If Chellas had in mind something similar to Chellas’ stit when he made up his $\Delta_a \varphi$ operator, he would have constrained the instigative alternatives (or actional alternatives) such that $R^a_t(h, h')$ only if $h \equiv_t h'$.

Still, it does not mean that $[a \text{cstit}: \varphi]_1$ is $\Delta_a \varphi$ without nuance. Our definition also suffers the fact that Chellas did not impose a “future branching only” [Che92, p. 489] nature of time and the independence of agents, while we inherit them from BT + AC structures.

6 Duration of an activity

Now, those operators indexed with a natural number $k$ may seem odd. But this is not odder than an iterated operator ‘next’ permitting to jump from instant to instant along a history. This is actually interesting to see what is going on if we allow such an operator:

$$\mathcal{M}, w/h \models X \varphi \iff \exists \bar{w} w < w', \exists w'' w < w'' < w', \text{ s.t. } \mathcal{M}, w'/h \models \varphi$$

In order to highlight how our primitive operators behave over time, it is easy to prove that $[a \text{cstit}: X^k \varphi] \leftrightarrow X^k[a \text{cstit}: \varphi]_k$, and $\square X^k \varphi \leftrightarrow X^k \square k \varphi$.

Let us designate a chain as being a set of linearly ordered moments. “In branching time, chains represent certain complex concrete events” [BPX01, p. 181].

While in the original STIT theory the $[cstit:]$ permits to express that an agent selects some set of histories (unbounded sets of ordered moments), underlying events are loosely characterized: they correspond to every chain we can construct on those histories. With $[cstit:]_k$ we clearly identify the set of events the agent has brought about: events composed of moments
between the moment of choice \( w \) and moments that are on the selected histories not farther than \( k \) instants after \( w \). We see that as a strength of the language.

**An example.** To give some intuition of possible applications of \( [acstit : \varphi]_k \) consider the following example. In an institutional context, it can be useful to reason about the length of an activity. For instance, given an operator for obligation \( \Box \), we could have a formula like

\[
phd(Mary) \rightarrow \Box [Mary\, \text{cstit}\,\,\text{Mary has written her thesis}]_{24}
\]

in the domain description, to state that a student can obtain a PhD only if he or she has achieved the writing of the thesis and has spent at least 24 months working on it.\(^2\) From Proposition 2, it indeed captures the notion of minimum. In such a modeling, it is like Mary chose at least 24 ‘clock ticks’ ago (that happen here to correspond to months) to write a thesis and it happens to have succeeded now.

### 7 Concluding remarks

The contribution here is humble: make clearer the link between logical operators by adding what can be seen conceptually harmless constraints in a discipline like computer science. First, it clearly highlights that the deliberative stit is a *local* achievement stit, or an achievement stit having the current moment as a witness. Second, it permits us to provide a more appropriate simulation of Chellas’ original operator of agency which was simply impossible without assuming discreteness.

**Acknowledgment**

I am debtful to Laure Vieu for her crucial observations on NSTIT and to Jan Broersen for pointing to me a problem in a preliminary version. I would also like to thank anonymous reviewers of this ESSLLI Student Session who did particularly relevant remarks regarding this paper.

**Bibliography**


\(^2\) In France a minimum of 2 years is imposed.
Some clarifications in logics of agency


Some clarifications in logics of agency
Evaluating Answer Extraction for Why-QA using RST-annotated Wikipedia texts

Suzan Verberne
Department of Linguistics, Radboud University Nijmegen
s.verberne@let.ru.nl

Abstract. In this paper the research focus is on the task of answer extraction for why-questions. As opposed to techniques for factoid QA, finding answers to why-questions involves exploiting text structure. Therefore, we approach the answer extraction problem as a discourse analysis task, using Rhetorical Structure Theory (RST) as framework. We evaluated this method using a set of why-questions that have been asked to the online question answering system answers.com with a corpus of answer fragments from Wikipedia, manually annotated with RST structures. The maximum recall that can be obtained by our answer extraction procedure is about 60%. We suggest paragraph retrieval as supplementary and alternative approach to RST-based answer extraction.

1 Introduction

In my PhD research project, I aim at developing a system for answering why-questions (why-QA). More specifically, I focus on the role that linguistic information and analysis can play in the process of why-QA.

In this paper the research focus is on the task of answer extraction for why-questions. In approaches to question answering (QA) involving factoid questions, named entity recognition can make a substantial contribution to identifying potential answers in a source document. For why-QA on the other hand, more sophisticated techniques are needed, because most answers consist of some kind of reasoning that cannot be expressed in a noun phrase. Arguments may be distributed over several sentences, making it necessary to exploit text structure. Therefore, we decided to approach the answer extraction problem as a discourse analysis task. We aim to find out to what extent discourse structure enables why-QA.

We created a system that uses discourse structure for answer extraction. In [13], we evaluated our approach using a set of elicited questions to a closed corpus (the RST Treebank [2]), with a moderate degree of success.
We hypothesized that part of the unsolved problems were due to the effect of the elicitation process: subjects might have been tempted to ‘invent’ why-questions that do not address the type of argumentation that one would expect for natural why-questions. Therefore, in the current paper, we aim to find out what the performance of discourse-based answer extraction is for why-questions that originate from real users’ information needs. To this end, we created a corpus consisting of why-questions asked to the online QA system answers.com, and a set of manually selected Wikipedia fragments which we annotated with discourse structure.

2 Answer extraction using discourse structure

As a model for discourse annotation, we use Rhetorical Structure Theory (RST), originally developed by Mann and Thompson [5] and adapted by Carlson et al. [2]. In RST, the smallest units of discourse are called elementary discourse units (EDUs). In terms of the RST model, a rhetorical relation typically holds between two EDUs, one of which (the nucleus) is more essential for conveying the propositional content than the other (the satellite). If two related EDUs are of equal importance, there is a multinuclear relation between them. Two or more related EDUs can be grouped together in a larger text span, which in its turn can participate in another relation. By grouping and relating spans of text, a hierarchical structure of the text is created. The main reason for using RST in the variant of Carlson et al. is that their rules and guidelines for segmenting discourse units and selecting relations are largely syntax-based, which fits the linguistic perspective of the current research. Moreover, Carlson et al. created a treebank of manually annotated Wall Street Journal texts with RST structures (the RST Treebank).

We presented our discourse-based approach to answer extraction in [13]. Our method is based on the idea that the topic of a why-question\(^1\) and its answer are siblings in the RST structure of the document, connected by a relation that is relevant for why-questions.

We performed two experiments for testing our method: (1) a manual analysis procedure and (2) and implementation of our approach.

First, we studied the theoretical upper bound of the contribution of RST to answer extraction by manually analyzing each question in our data collection and its corresponding RST structure. We apply the following manual analysis steps to each of the question-answer pairs:

1. Identify the topic of the question; in the RST tree of the source document, identify the span(s) of text that express(es) the same proposition as the question topic;

\(^1\)The topic of a why-question is the proposition that is questioned. A why-question has the form ‘WHY P?’, in which the proposition P is the topic. [10]
2. Does the topic span participate in an RST relation? If it does, select the span (nucleus or satellite) that participates, and take note of the relation type;

3. Select the topic span’s sibling as a potential answer;

4. Decide whether this span is satisfactory as answer to the question.

The effects of this procedure can best be demonstrated by means of an example. Consider the question Why is the funny bone so called? The following text fragment contains the answer:

“The ulnar nerve comes from the medial cord of the brachial plexus, and runs inferior on the medial/posterior aspect of the humerus down the arm, going behind the medial epicondyle at the elbow. Because of the mild pain and tingling throughout the forearm associated with sudden compression of the nerve at this point, it is sometimes called the funny bone. (It may also have to do with its location relative to the humerus, as well as the fact that ‘humerus’ is homophonic to the word ‘humorous’).”

In this text, we identify the following clause representing the question topic: it is sometimes called the funny bone. In figure 1 below the RST annotation of the paragraph is shown. Here we see that the span representing the question topic is EDU number 6, which is the nucleus of an explanation-argumentative relation.

![RST Structure](image)

Figure 1.1: Part of the RST structure for the answer paragraph on the funny bone.

The sibling of the topic span is span 4-5 in the hierarchy: Because of the mild pain and tingling throughout the forearm associated with sudden
Evaluating Answer Extraction for Why-QA using RST-annotated Wikipedia texts

compression of the nerve at this point. We judge this span as a satisfactory answer to the question. However, we also note that the complete paragraph would have been a more complete (and therefore better) answer than this single clause, because the sentence contains one broken anaphoric reference (this point) and lacks background information on the nerve mentioned.

The second experiment is the implementation of an algorithm in Perl that reflects the manual analysis steps described above. We built an indexing script that takes as input file the RST structure of a document, and searches it for instances of potentially relevant why-relations. It then extracts for each relation both the participating spans and its relation type and saves the information to an index file (in plain text).

For the actual retrieval task, we wrote a second Perl script that takes as input one of the document indices, and a question related to the document. Then it performs the following steps:

1. Read the index file, normalize and lemmatize each span in the index;

2. Read the question, normalize and lemmatize it;

3. For each span in the index, calculate its likelihood using a probability model that takes into account its lexical overlap with the question and a prior on the relation type it participates in.

4. Save all spans with a likelihood greater than a predefined threshold and rank the spans according to their likelihood;

5. Retrieve as potential answers the siblings of each of these spans.

In [13], we created a test corpus consisting of seven texts from the RST Treebank and 372 why-questions elicited from native speakers to these documents. We performed both experiments (manual analysis and implementation) on this data collection. Following our manual analysis procedure (first experiment), we found a satisfactory answer for 58.0% of the questions. Thus, we argued that the maximum recall that can be achieved using our discourse-based answer extraction approach is 58.0%. The implementation of our algorithm (second experiment) reaches a recall of 53.3% with a mean reciprocal rank of 0.662.

We consider a recall of 53.3% (and a maximum recall of 58.0%) as mediocre at best. An in-depth analysis of the questions for which the answers could not be found suggested that a fair proportion of the questions were somewhat artificial, probably invented for the purpose of the experiment. Thus, the elicitation procedure may result in a set of questions that are not representative for users’ actual information needs.
3 Real users’ why-questions and answer fragments

In order to test the discourse-based procedure for answer extraction on a question set that is more representative for questions asked to a QA system, we created a data set from questions that have been posed to the online domain-independent QA system answers.com. Hovy et al. downloaded 17,000 questions from answers.com for their Webclpedia collection [4]. 805 questions from the Webclpedia set are why-questions—pragmatically defined as questions starting with the word why. The source of these questions guarantees that they originate from users’ information needs. We randomly selected 400 of these why-questions for our data collection.

We first study these 400 why-questions from the Webclpedia set independently from their answer documents. In [12], we created a typology of why-questions based on the classification of adverbial clauses by Quirk et al. [8]. We originally chose four classes for the semantic answer type of why-questions: ‘motivation’, ‘cause’, ‘circumstance’ and ‘generic purpose’. Of these, cause (52%) and motivation (37%) were by far the most frequent answer types in our set of elicited why-questions pertaining to newspaper texts [12]. From other research reported on in the literature it appears that knowing the answer type helps a QA system in selecting potential answers. Some work that we did on answer type prediction is reported on in [11].

For the current set of Webclpedia why-questions, we find that the proportion of questions expecting a motivation as answer is much smaller than for the elicited questions (10%), and that ‘circumstance’ and ‘generic purpose’ are again negligible as question classes. The remaining category, ‘cause’, is too general as a class for all other questions. Therefore, we decide to split the current collection of Webclpedia why-questions into five classes:

- Motivation (10%), for example: Why did NBC reject the first “Star Trek” episode, “The Cage” in 1965?
- Physical Explanation (42%), for example: Why can’t people sneeze with their eyes open?
- Non-physical explanation (30%), for example: Why is the color purple associated with royalty?
- Etymology (12%), for example: Why are chicken wings called Buffalo wings?
- Trivial/Nonsense (6%), for example: Why is the word “abbreviation” so long?

For analysis and development purposes, we created a set of answer fragments to the 400 Webclpedia why-questions. We manually extracted these fragments from Wikipedia using Google’s domain search on en.wikipedia.org.

259
We chose Wikipedia as source for several reasons: it is relatively stable compared to the Internet as a whole, and its content is reliable and accurate [3]. For 54% of the questions, we can find the answer in Wikipedia. Of the other 46%, some questions had false question propositions (e.g. Why is a computer cabinet always white?) and other questions seem to be either too specific (e.g. Why do cows lie down before it rains?) or too trivial (e.g. Why is weird spelled w-e-i-r-d and not w-i-e-r-d?) for Wikipedia to contain the answer. In a large majority of cases (94%) the length of the answer does not exceed a single paragraph.

We let two experienced annotators create RST structures for the 216 answer fragments from Wikipedia. For answer fragments shorter than one paragraph, we selected the complete paragraph for annotation. We also added the previous paragraph or the section heading to the fragment if these provided essential information for understanding the paragraph containing the answer. We did not inform the annotators about the purpose of their annotations.

For determining the consistency of our annotations, we measure inter-annotator agreement. We let both annotators annotate the first ten fragments from our data set, and we calculate $\kappa$ scores for both segmentation and hierarchy (nuclearity). For the calculation of $\kappa$, we follow Marcu’s [6] definition of $\kappa_u$ for segmentation and $\kappa_n$ for nuclearity. We get a moderate agreement for segmentation ($\kappa = 0.54$) and low agreement for nuclearity ($\kappa = 0.13$). Marcu et al. found 0.72 and 0.67 respectively for $\kappa_u$ and $\kappa_n$ for their RST Treebank, which is much higher. We assume that the difference in $\kappa$ scores is due to the procedure used to obtain the annotations: Marcu et al. trained their annotators elaborately for the purpose of maximizing the consistency of the annotations. In our project we have to rely on annotators who received substantial training in applying RST, but they were, due to temporal and financial limitations, never put in a situation where they had to reach a common interpretation of a set of training texts. Despite the low agreement for the nuclearity annotations, we still decide to press forward and use the resulting annotations for extracting answers for why-questions.

We now have a set of why-questions and answers that differs from the first data collection in (a) the source of the questions (real user questions instead of elicited questions), (b) the source of the answer corpus (newly annotated encyclopedia fragments instead of pre-annotated newspaper texts), and (c) the collection procedure (answers extracted for existing questions instead of questions formulated for existing answer documents).

4 Results and discussion

We executed the two experiments described in section 2 on our Webclopedia/Wikipedia collection, following the same procedures as for the collection
of elicited questions. We only considered the questions for which we were able to find an answer in Wikipedia (54% of all questions).

In the first experiment, involving manual analysis, we find that our answer extraction procedure leads to a satisfactory answer for 60.6% of our Webclopedia questions. The remaining 39.4% suffers from one of the following problems (in the order of the analysis steps):

1. The question topic is not represented by a text span in the answer fragment (18% of all questions);
2. The text span representing the question topic does not participate in an RST relation (2%);
3. The sibling of the span representing the question topic is not a satisfactory answer (21%).

In the last case, the correct answer is somewhere else in the tree or in the same discourse unit as the question topic. For example, the clause *Buffalo wings are named after the city of Buffalo, New York* contains both the question topic *chicken wings are called Buffalo wings* and its answer.

We find no significant differences in success rate for the fragments that were annotated by the different annotators. This suggests that the low inter-annotator agreement has no noticeable influence on the answer extraction task that we consider. This may be because the majority of the RST relations that are relevant for *why*-QA are so obvious that annotators are likely to treat these similarly, but this remains to be seen.

If we compare the success rate of the proposed answer extraction procedure for the current data collection to the success rate that we found for the elicited questions with the RST Treebank (as described in section 2), we see highly similar results: for the Webclopedia questions, 60.6% of answers can be found through manual topic matching and sibling selection. For the elicited questions, this figure was 58.0%. Thus, although the questions in both data collections came from different sources, our answer selection procedure showed highly similar results for both sets.

We also compare the set of relation types addressed by the Webclopedia questions to the set of relation types addressed by the elicited questions. Table 1 gives an overview of the relation types that leads to the correct answer for at least 6% of the questions where our discourse-based answer extraction approach succeeds in either the Webclopedia set or the elicitation data. In the second and third column are the figures for the RST Treebank and the corresponding elicited questions. In columns four and five are the percentages for the Wikipedia corpus and the Webclopedia questions. We see for example that 18.0% of relations in the RST Treebank are elaboration relations, and for 27.0% of *why*-questions where our approach succeeds, it is an elaboration relation that connects question topic and answer. For the Wikipedia corpus, these numbers are 22.4% and 20.8% respectively.
Table 1.1: Distribution of relation types in corpora and question sets

<table>
<thead>
<tr>
<th>Relation type</th>
<th>% of relations</th>
<th>% of questions</th>
<th>% of relations</th>
<th>% of questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elaboration</td>
<td>18.0%</td>
<td>27.0%</td>
<td>22.4%</td>
<td>20.8%</td>
</tr>
<tr>
<td>Explanation</td>
<td>1.4%</td>
<td>7.1%</td>
<td>3.5%</td>
<td>20.0%</td>
</tr>
<tr>
<td>Circumstance</td>
<td>1.7%</td>
<td>0.5%</td>
<td>8.1%</td>
<td>16.0%</td>
</tr>
<tr>
<td>Background</td>
<td>0.5%</td>
<td>0.0%</td>
<td>4.3%</td>
<td>8.8%</td>
</tr>
<tr>
<td>Purpose</td>
<td>1.3%</td>
<td>14.3%</td>
<td>2.6%</td>
<td>7.2%</td>
</tr>
<tr>
<td>Consequence</td>
<td>1.0%</td>
<td>15.3%</td>
<td>0.8%</td>
<td>2.4%</td>
</tr>
<tr>
<td>Reason</td>
<td>0.6%</td>
<td>9.7%</td>
<td>0.9%</td>
<td>4.0%</td>
</tr>
<tr>
<td>Result</td>
<td>0.7%</td>
<td>9.7%</td>
<td>1.2%</td>
<td>2.4%</td>
</tr>
</tbody>
</table>

Although the success rate of our discourse-based answer extraction approach is similar for the Webclopedia and elicitation data collections (around 60%), we see some interesting differences between the two data collections in table 1. First, some relations differ considerably in their relative frequencies in both corpora (columns 2 and 4): explanation-argumentative (1.4% versus 3.5%), circumstance (1.7% versus 8.1%) and background (0.5% versus 4.3%). These differences are partly due to differences in annotation styles, and partly the result of differences in text types: the RST Treebank contains newspaper texts whereas the Wikipedia corpus consists of encyclopedic items where one would expect a higher density of explanations.

Secondly, we see large differences between the relative frequencies of the relations in the set of questions (columns 3 and 5). Again, the main differences lie in the relation types explanation-argumentative, circumstance and background, but also purpose, consequence, reason and result show large differences. The last four are the most interesting since the relative frequencies of these relations are more similar for the two source corpora than for their question sets. This means that the differences for these relation types come from the question source: questions asked to a QA system are apparently more likely to expect explanations, backgrounds and circumstances as answer than elicited questions. Elicited questions on the other hand refer to purposes, consequences and reasons more often. This matches to the differences in semantic answer types that we saw in section 3.1 if we take into account that purpose and reason, as defined by Carlson et al. [1], correspond to our definition of the answer type motivation [13].

The RST relations most frequently addressed in our Webclopedia question set are elaboration, explanation-argumentative, circumstance, background and purpose. Here, we see that the very general relation type ela-
boration is the most frequently occurring relation type for why-questions. However, there is a relatively small proportion of the question topics that participate in an elaboration relation for which this relation leads to a satisfactory answer: 49%. In other words: the predictive power of elaboration relations for why-questions is small. The predictive power for the question topics participating in an explanation-argumentative relation is much larger: for 89% of the question topics that participate in an explanation-argumentative relation, this relation leads to a satisfactory answer. For the question topics participating in a circumstance, background and purpose relation, these relations lead to a satisfactory answer in 77%, 85% and 100% of participating question topics respectively. Thus, we can conclude that the relation types explanation-argumentative, circumstance, background and purpose are valuable for finding answers to why-questions, whereas elaboration relations have low relevance. Furthermore, the predictive power of some types of RST relations confirms the expected importance of answer type determination. If we can predict the answer type from the question, and we know which RST relations represent this answer type, then we can apply the knowledge on the expected answer type for answer selection and ranking.

Our manual analyses described above lead to the conclusion that the maximum recall that can be achieved using our discourse-based answer extraction approach is around 60%. The success rate that we found is similar for both data collections, but the relation types addressed are different for the two corpora.

We then performed the second experiment, implementation of our algorithm, to the Webclopedia/Wikipedia data collection. Here, we found large differences between the two data collections: our implementation obtains a recall of only 25.9% on the Webclopedia/Wikipedia data set, whereas it had scored 53.3% for the elicited questions. This difference comes from the third step of our algorithm: matching the question topic to spans in the source text using lexical overlap measures. Questions elicited from subjects who are reading a text tend to use the same terms as those in the texts. This suggests that the results obtained using the Wall Street Journal texts do not generalize to any other setting. For the Webclopedia questions, lexical overlap is much smaller because these questions were formulated completely independently from a specific text. Assuming that the Webclopedia/Wikipedia set is representative to an actual question answering setting, we should acknowledge the problem of small lexical overlap between question and source document in the system under development.

5 Conclusions and further research

We created a corpus of why-questions consisting of 400 questions from the Webclopedia question set and corresponding answer fragments from
Wikipedia, manually annotated with RST relations. This data collection may be of interest for other researchers in the field of question answering or discourse analysis.\footnote{We have made both our data collections available through the project’s web site \url{http://lands.let.ru.nl/~sverbern/}}

We evaluated an answer extraction method for why-questions based on the idea that question topic and answer are siblings in the RST structure. We found that our procedure is potentially successful for 60% of why-questions. The current implementation of our procedure can retrieve 25.9% of the manually selected answers to the Webclopedia questions from the corresponding Wikipedia document.

We conclude that discourse structure can be useful in solving at least a subset of why-questions and that some relation types have a predictive power in answer selection. However, our answer extraction approach should be combined with other methods in order to increase recall.

We consider paragraph retrieval as alternative and supplementary approach. We found that for 44% of the cases where the procedure succeeds, the complete answer paragraph would (in our judgement) be a better answer to the question than the answer span in the RST tree only. Moreover, for 30% of the questions for which the procedure does not succeed (because the question topic is not in the text or question topic and answer are no siblings), the complete paragraph gives the answer. Thus, paragraph retrieval is a good additive solution to discourse-based answer extraction. Since some types of RST relations appears to have a high predictive power in answer selection, we aim at developing a method for paragraph retrieval in which we incorporate knowledge about the presence of relevant RST relations.

We also plan to perform user studies in order to determine what answer form users prefer for different types of why-questions and answers. This way, we aim to find out whether paragraph retrieval with information from (partial) RST annotations can be a good alternative to the strict procedure of topic matching and sibling selection.

We should also note that in a future application of why-QA using RST, the system will not have access to a manually annotated corpus—it has to deal with automatically annotated data. We assume that automatic RST annotations will be less complete and less precise than the manual annotations are. Some work has been done on automatically annotating text with discourse structure. Promising in this direction is the done work by Marcu and Echihabi\cite{Marcu:2002} and Soricut and Marcu\cite{Soricut:2005}. We plan to investigate to what extent we can achieve automatic partial discourse annotations that are specifically equipped to finding answers to why-questions.

\footnote{We have made both our data collections available through the project’s web site \url{http://lands.let.ru.nl/~sverbern/}}
Bibliography


