Re-ranking based on Syntactic Dependencies in Prior-Art Retrieval

Eva D'hondt, Suzan Verberne, Nelleke Oostdijk, Lou Boves
Centre for Language Studies & Information Foraging Lab,
Radboud University Nijmegen,
(e.dhondt|s.verberne|n.oostdijk|l.boves)@let.ru.nl

ABSTRACT

In this paper we present an experiment using syntax (in the form of dependency triplets) to rerank retrieval results in the patent domain. This work is a follow-up experiment of our participation in the first CLEF-IP track, which focussed on prior art retrieval. We shall first describe the work done in our participation to the CLEF-IP track and then go on to show why improving Mean Average Precision (MAP) is important to the patent searchers community. We then introduce an additional reranking step to our BOW retrieval approach which is based on syntactic information. Using syntactic structures called Dependency Triplets as index terms we perform a second retrieval step within the retrieved result sets and examine if the ranking of the relevant documents (captured by the MAP score) can be improved for prior art search.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search

General Terms

Dependency Triples

Keywords

Prior Art, patent retrieval, syntactic units

1. INTRODUCTION

Patent retrieval is a rising research topic in the western Information Retrieval (IR) community. Though it already was the topic of workshops in SIGIR 2000 and ACL 2003 and has been a recurring track in the NTCIR workshops since 2002, it has not gathered a lot of attention from the western Information Retrieval community, mainly because the document collections of the NTCIR workshops are more focussed on Asian languages. In 2009, however, the first Patent Retrieval track with a focus on European languages (CLEF-IP)[1] was organized by the Information Retrieval Facility (IFR) as part of the CLEF 2009 evaluation campaign.[2] The general aim of the track is to explore patent searching as an information retrieval task and bridge the gap between the IR community and the world of professional patent search.

The goal of the 2009 CLEF-IP track was 'to find patent documents[3] that constitute prior art[4] to a given patent' [20]. In this retrieval task each topic query was a (partial) patent document which could be used as one long query or from which smaller queries could be generated. The track featured two kinds of tasks: In the Main Task prior art had to be found in any one (or combination) of the three following languages: English, French and German; three optional subtasks used parallel monolingual topics in one of the three languages. In total 15 European teams participated in the track. Because of this high participation rate, the CLEF-IP track will be sure to continue next year.

At the Radboud University of Nijmegen we decided to participate in the CLEF-IP track because it is related to the focus of the Text Mining for Intellectual Property (TM4IP) project[15] that we are currently carrying out. In this project we investigate how linguistic knowledge can be used effectively to improve the retrieval process and facilitate interactive search for patent retrieval. Because the task of prior-art retrieval was new to us, we chose to implement a baseline approach to investigate how well traditional IR techniques work for this type of data and where improvements would be most effective. These results will effectively serve as a baseline for further experiments as we explore the influence of using dependency triplets[5] for various IR tasks on the same patent corpus.

[1] http://www.ir-facility.org/research/evaluation/clef-ip-09/overview

[2] See http://www.clef-campaign.org

[3] In this paper we use the following terminology: a 'patent document' is physical document which is a version of a patent (application) at a certain point in time; A 'patent' is a set of documents that carry the same patentID code. This is explained in more detail in section 3.1.

[4] Prior art for a patent (application) means any document (mostly legal or scientific) that was published before the filing date of the patent and which describes the same or a similar invention.

[5] A dependency triplet is a unit that consists of two open category words and a meaningful grammatical relation that binds them.

In the CLEF-IP task we used a standard retrieval approach based on keyword matching, using the Lemur retrieval engine and the TF-IDF ranking algorithm. This baseline run achieved moderate results compared to the other participants (Recall@100= 0.22 and MAP=0.054). Overall, the results of all participants were rather low, compared to retrieval results in other tasks: Recall@100 ranged from 0.58 to 0.02 and Mean Average Precision (MAP)[6] from 0.11 to 0.00 (with one outlier: the run submitted by the Humboldt University which achieved 0.27). These general results will be further discussed in section 2.4.

The MAP score and the Recall score are the two most important measures for patent retrieval [3]. Recall must be very high, because for patent searchers it is extremely important to find ALL relevant documents. The financial repercussions of an incomplete prior art search can be severe, even if the patent has already been granted. But while recall is important, it is also clear that patent searchers cannot afford to process large result sets comprising thousands of patents that have to be browsed through completely: Patent retrieval is a highly interactive search task where the information need is constantly modified throughout the search. Finding a particularly relevant document at an early stage of the search will enhance the effectiveness of the remainder of the search task. (For example, by adding new keywords, IPC[7] codes, etc. gained from this document to the query.) Therefore, improving the ranking of the relevant documents in the result set is important to the patent searcher.

There is evidence in the IR literature (see section 2.1) that using dependency relations to rerank a small, already retrieved set of documents can be very successful for ad-hoc document retrieval and QA. The dependency model used in the TM4IP project differs from most other dependency models in that it is developed for IR purposes and is therefore linguistically less detailed than other models. In the project we are currently developing the AEGIR parser, a rule-based dependency parser which is geared towards the specifics of the language used in patents and which is more robust than other general-language parsers. This parser generates dependency triplets from the input text, which are then -in turn- used as index terms in the interactive retrieval system (also under development).

In this paper we focus on improving MAP of the result list, produced in the CLEF-IP experiment, by adding an extra step. To this end we perform an additional reranking operation on the result set using syntactic information in the form of dependency triplets[8]

2. BACKGROUND

2.1 Syntax-based retrieval

In Information Retrieval the bag-of-words approach (BOW) is the approach most frequently used for all types of IR tasks. It is attractive to researchers because it makes the model simple, easily manageable and comprehensible. However, a recurring criticism on the BOW approach is the fact that by splitting the text up into single terms, the model does not take into account the immediate context of the terms and subsequent relations between terms. For example, a simple BOW-based retrieval system cannot differentiate between the following two queries: *bank terminology* and *terminology bank* [29]

In the last two decades, several approaches have been developed that use larger retrieval units, namely phrases. Phrases can be defined by their statistical properties or syntactic characteristics or a combination of two. The most successful statistical approaches are proximity-based phrase indexing [10], the n-gram retrieval model[9] [24] and the term dependency modelling approach [11], [18]. These approaches focus on taking context into account and are able to capture some (dependency) relations between terms on the basis of their collocation frequencies. However, they typically fail on long distance dependencies.

The more linguistically-motivated approaches, such as [23], [26], [2] have focussed on extracting syntactic units from the text using linguistic information. These phrases can either take the form of a head-modifier pair or a (partial) dependency tree. Several studies have investigated the effect of using syntactic versus statistical phrases as index terms: [10],[16], [13], [1] found that there is only a small improvement when syntactic relations are taken into account in the retrieval process. Syntactic phrases have been found to be useful, however, for improving the ranking of the results found by a BOW approach, at least for ad-hoc search [4] and QA [8],[28]. [6] reports that the longer the queries, the more useful NLP techniques like extracting dependency pairs can become, though he adds that (at least for ad hoc search) the benefit is limited.

[25] argues that one of the reasons for the disappointing results in dependency-based retrieval could be the fact that the earlier systems did not take the *variability of the structure* of the syntactic phrase into account: In a noun phrase like *World Bank criticism* a syntactic phrase that contains a compound like 'World Bank' is a much more important retrieval unit than *Bank criticism* and should be given more weight as an index term. [19] remarks that part of the discouraging effect of phrases in text retrieval stems from the fact that they must be normalized to a standard form (in order to rise above syntactic and lexical variation). Such transformations are complex and prone to errors. The removal of function words (e.g. prepositions, determiners, ..) plays an important role in this normalisation process [10].

2.2 Syntax in Patent Retrieval

The majority of the search engines used by the patent search community today are keyword-based, using a general-purpose text search engine. Academic research on patent retrieval has mainly been focussed on the relative weighing of the index terms [17] and on exploiting the patent document structure to boost retrieval [17]. There is a lot of attention for query reformulation at the moment, as could be seen in the

[6]MAP is a measure of how high the relevant documents appear in the result list, measured over all queries.

[7]The International Patent Classification, used by all major patent offices.

[8]In our approach to dependency triplets we consider them as single index units, not as relationships between two separate index terms.

[9]For an overview of related articles and patents, see http://www.cs.umbc.edu/ngram/

CLEF-IP track where 5 out of 14 teams actively explored different query term selection and query reformulation strategies. For an overview of the state of the art in academic and commercial systems, see [5].

There are not many approaches in the patent domain that use syntactic phrases or structures comparable to our approach which we will explain in section 2.3. Systems like [9] and [7] perform a combination of syntactic and semantic analysis on the documents and use the results to generate concept units. The only purely syntactic approach is [21], who uses deep linguistic analysis in the form of predicate-argument analysis (implying semantic role labelling) to improve readability of the claims section. Her system is the first step in a suggested patent summarization method.

2.3 The CLEF-IP track

In answer to a growing demand from the patent searcher community for reliable and improved patent search engines the first CLEF-IP track was organised by the IRF. As was explained in section 1, it aims to bring the IR community and the world of professional patent search closer together to create new and innovative retrieval systems. The first track can be considered a major success as it received a lot of interest from the IR community and –in turn– presented the IR community with a patent corpus of significant size within an integrated and single IR evaluation collection. The results of the participating groups in the patent track yielded some interesting insights into the particulars of patent retrieval: as mentioned above the overall precision and recall results in this task were quite low (average Precision@100= 0.02, average Recall@100=0.38, average MAP = 0.07, except for one outlier) compared to the results in other retrieval tracks.

There are a number of reasons for these low scores: First of all, some of the documents were 'unfindable': 17% of the patent documents in the collection contained so little information, e.g. only the title which is poorly informative for patent retrieval [27], that they could not be retrieved. Secondly, the relevance assessments were based on search reports and the citations in the original patent only. This means that they were conceptually-based and not text-based and may therefore have been too limited[10]. Finally, in order to perform retrieval on the patent level, instead of the document level, some of the participating groups created 'virtual patents': for each field in the patent the most recent information was selected from one of the documents with that patentID. These fields were glued together to form one whole 'virtual' patent. It is, however, not necessarily true that the most recent fields are the most informative [27]. This selection operation may have resulted in a loss of information. However, even without these impediments, it is clear that patent retrieval is a difficult task for standard retrieval methods.

2.4 The TM4IP project

At the Radboud University Nijmegen, we are currently involved in the Text Mining for Intellectual Property (TM4IP) project[15], which is directed at developing an approach to interactive patent search using syntactic structures in the form of dependency triplets as search terms and for computing the relevance ranking. While the idea of using (partial) syntactic phrases as index terms is not new (see section 2.1), the dependency model used in TM4IP differs from previous attempts in that it is based on the notion of *aboutness* to suit retrieval purposes. Aboutness is a difficult concept to define and has many different interpretations in the literature. In IR it is defined as follows: the user of a retrieval system expects the system, in response to a query, to supply a list of documents which are *about* that query. Practical retrieval systems using single words as terms are based on an extremely simpleminded notion of aboutness. For our system, the concept of *aboutness* implies that we do not allow any words in the dependency triplets that have no classificatory value as keywords (by themselves) [15].

In this project a rule-based dependency parser has been constructed that is now being tuned to deal with English technical texts. In the near future, this parser will also incorporate frequency information on words and on triplets and will thus become a hybrid parser. This parser generates dependency triplets (structured units, containing word forms and dependency relations) from the input text, which are then –in turn– used as index terms in the retrieval system. The aim of the project is to successfully use linguistic knowledge (in the form of dependency triplets) to improve the retrieval process and facilitate interactive search for patent retrieval. We have already achieved good results using the dependency triplets as basic units for the classifier for patent documents that is also a part of our system [14]. Using dependency triplets as classification terms, we reached a high accuracy in the (pre)classification of patent applications in their correct IPC classes.

The full dependency triplet-based patent search system is still under development. Therefore, in this paper we investigate the effect of using dependency triplets for improving the relevance ranking of documents that have been retrieved by some conventional search engine. Literature shows that re-ranking with dependency triplets can be successful (see section 2.1).

3. METHODOLOGY

3.1 Data

The CLEF-IP corpus consists of European Patent Office (EPO) documents that have been published between 1985 and 2000, covering English, French, and German patents. In total, the corpus contains 1,958,955 patent-documents pertaining to 1,022,388 patents (75GB) as one patent can consist of multiple XML files: A patent can consist of several documents that were produced at different stages of a patent realization.[11] For example, a so-called A2 document (the patent application in its barest form, submitted at the beginning of the patent application process) can contain only a title and perhaps an abstract, while a B1 document (a granted patent, usually finished three years after the initial application) will contain a title, abstract, claims and description section.

The heterogenity of the corpus has certain implications for

[10]http://www.clef-campaign.org/2009/working_notes/CLEF-2009WNContents.html

[11]For an overview of the patent kind codes used in the corpus, see http://www.delphion.com/help/kindcodes under EPO.

the search process: While it seems preferable to search only in the B1 documents, this would exclude a large number of documents from the search that could be relevant while searching for prior art: some patents only consist of an A2 document.

In the CLEF-IP 2009 track the participating teams were provided with 4 different sets of topics (S,M,L,XL). We opted to do runs on the smallest set (the S data set) for both the Main and the English task. This set contained 500 topics. Because the information in these topics was different for both tasks[12] we focussed on the data that was available in all the topics: the English claims sections. As only 70% of the CLEF-IP corpus contained English claims, this means that a substantial part of the corpus could not be retrieved.[13] By reducing the patent documents to the claims sections only, we gained consistency (all the documents to be retrieved have the same style of writing and are not empty). Even so, improving consistency in the way we did comes with a price. We might have thrown away that part of the document containing the relevant information. In the patent retrieval literature, however, there is evidence [12],[22] that the claims section is the more informative part of the patent document. Nonetheless, we may wonder if –for the reranking experiment– limiting our document set to claims text only will not have an adverse effect on the generation of the index terms (dependency triplets): It might be that this will put an additional strain on the parser, as the language in claims is notoriously difficult to read and highly complex, therefore quite difficult to parse correctly.

3.2 Baseline approach

3.2.1 *Queries*

After removing punctuation and stopwords we took all remaining words in the claims section together as one long query (weighted in retrieval with TF-IDF). No stemming was conducted.

3.2.2 *Indexing and Retrieval using Lemur*

We extracted the claims sections from all English patent documents in the corpus and removed all XML markup from the texts by means of a preprocessing script. Since there may be multiple documents that carry the same patent-ID, we concatenated the claims sections pertaining to one patent ID into one document in the index file. We saved all patent claims in the Lemur index format with the patent IDs as DOCIDs. They were then indexed using the BuildIndex function of Lemur with the indri IndexType and a stop word list for general English. The batch retrieval was then performed using TF-IDF.

3.3 Reranking experiment

3.3.1 *Data selection*

[12]Some of the topics for the Main Task contained the abstract content as well as the full information of the granted patent except for citation information, while the topic patents for the English Task only contained the title and claims elements of the granted patent [3].

[13]Of the 30% percent that could not be retrieved by our system, 7% were documents that only had claims in German or French but not in English, 6% only contained a title and abstract, usually in English and 17% only contained a title.

In the baseline experiment we retrieved 100 results for each of the 500 topics but because some documents were attributed to multiple topics we only retrieved a total of 39,802 unique documents. In total the retrieved documents contained around 52 million words. The average sentence length in these document was 49 words and the longest sentence in the retrieved documents consisted of 451 words.

In the reranking experiment we took all 100 documents of the result set (per topic), parsed them (see 3.3.2), used Lemur to create a separate index containing all the triplets of the retrieved documents per topic and performed a second retrieval on these indices. On average, the result sets contained around 85,000 words each.

We chose this set-up to compare the impact of dependency triplets in the ranking of the documents. For each topic, the exact same hundred documents are available in the index that were found (for that topic) in the baseline experiment. Consequently, the same (number of) relevant documents will be found in the second retrieval step. Therefore, recall and precision will remain the same in the second experiment and only the ranking of the (relevant) documents (measured in MAP) can be subject to change.

3.3.2 *Pre-processing*

We parsed the topics and the 39,802 retrieved documents of the CLEF-IP corpus using the AEGIR parser (version 1.1). The grammar from which the parser was generated comprises some 200 rules. The dependency model used by the parser has the following format: `[term1, relator, term2]`. The sentence 'The system consists of four separate modules' will be turned into the following triplets: `[system,SUBJ,consists]`, `[consists,PREPof,modules]`, `[modules,ATTR,separate]`, `[modules,QUANT,four]`. Our dependency model is based on the notion of aboutness: with a few exceptions only open category members are allowed as head or modifier. In the example given above, the determiner 'the' is not allowed into the triplets. We used a small set of relators which mirror basic semantic relations :

- SUBJ(ect):
  - 'The method describes' `[method,SUBJ,describes]`;
  - '(Object) claimed by Microsoft' `[Microsoft,SUBJ,claimed]`;
- OBJ(ect) :
  - '(I) killed the man' `[killed,OBJ,man]` ;
  - 'The air is compressed (by subject)' `[compressed,OBJ,air]` ;
- ATTR(ibutive):
  - 'the smaller wheel' `[wheel,ATTR,smaller]`;
- PRED(icate):
  - 'the element is uranium'

[element,PRED,uranium];

- MOD(ifier):
  - 'very green'
    [green,MOD,very];
- QUANT(ifier):
  - 'four wheels'
    [wheels,QUANT,four];
- ... .

We did not apply any lemmatisation (or stemming) to the words in the triplets.

To limit the time needed to parse all 39,802 documents, we decided to introduce a maximal time limit for the parser (1800 seconds per parse). With this procedure two topic documents failed to parse, as well as a very small fraction of documents returned in the retrieval process (0.0025%). Though this may not seem much, it does mean that every time the parser failed, absolutely no triplets were generated for that portion of the text, which makes it invisible for the retrieval system in the reranking experiment. Numerically, however, these missing triplets are only a fraction of the corpus of some 32 million triplets that were generated. In total, it took a week to parse the topics and the documents in the result sets on a cluster of single core PCs, most of which had no more than 1 Gbyte of internal memory.

3.3.3 *Query and indexing*

Triplets from both the topic and result set documents were transformed into a single string using a perl script. For example, `[fact,ATTR,well-known]` and `[system,SUBJ,performs]` were transformed into `factattrwell_known` and `systemsubjperforms`, respectively. These strings then served as index and query terms for direct matching ('Bag of Triplets' matching). We constructed 500 separate indices (one per topic) using the BuildIndex function of Lemur with the indri IndexType. Each index contained the strings of those documents that were retrieved for that topic in the baseline run. For each of the 500 topic queries, batch retrieval was then performed on its specific index using the TF-IDF ranking algorithm.

Since we performed a second retrieval step, we take the risk of not re-retrieving a portion of the documents retrieved with the BOW retrieval. On average, we retrieved 90.1% of the 100 documents per topic. We identified the missing documents using a python script that compared the baseline result list with the reranking result list and added these missing documents to the end of the reranking result list in the relative order in which they had been found in the baseline.

The 498 successful[14] individual retrieval result sets of the reranking experiment were compared with the results from the baseline experiment using a python script in order to calculate the Precision, Recall and MAP measures. We also calculated the rank of the first relevant document per query.

[14]As mentioned above, two topic parses failed and therefore we could not compare the retrieval sets.

4. RESULTS

4.1 Baseline retrieval results

During the baseline experiment we retrieved a total of 645 relevant documents in the CLEF-IP corpus for the 500 topic documents. We achieved a score of 0.22 for recall and a MAP score of 0.054.

4.2 Reranking retrieval results

The reranking system performed significantly worse than the baseline system: The MAP score dropped from 0.054 for the baseline system to 0.045 for the reranking system. ($p < 0.001$ according to the Wilcoxon Signed Ranks Test).

Of the 645 relevant documents that were retrieved in both experiments, 8 had the same ranking in the baseline as in the reranking result set (1.3%). In 537 cases, the relevant document had a higher ranking in the baseline approach (83.3%) and only in 100 cases did the reranking approach produce better rankings for the documents (15.5%). On average, the documents either dropped 40 ranks in the reranking result set or rose 18 ranks compared to the baseline rankings.

4.3 Parser evaluation

The outcome of the retrieval process is highly dependent on the quality and quantity of the generated triplets. We therefore evaluated the accuracy of our parser on a small test set of 14 sentences (656 words) taken randomly from the claims sections. To create this test set two of the authors independently created dependency triplets for different parts of the test set. There was an overlap of 5 sentences, each of around 40 words, which was used to calculate inter-annotator agreement for the test set. The inter-annotator agreement was 74%[15], indicating substantial agreement on annotation. The language typically used in claims sections ('legalese') has – apart from other particularities – a lot of syntactic ambiguities and therefore it is not surprising that the biggest differences in manual annotations could be attributed to different interpretations of coordinations for the SUBJ relations and of PP attachments. The following sentence is an example of the first difficulty:

> 'The device claimed in claim 1 consists of .... [15 words] and uses 5 volt.'

As these dependencies can be stretched quite long (15 words in between), it is very difficult, even for a human, to see which word should be connected to the second verb. An example of the second problem can be seen in the following example:

> 'The mapper is adapted to divide a stream of bits from the encoder into at least a first period by the rightful application of ... ' .

This is a well-known problem for any parser, usually demonstrated with the famous 'I saw the man with the telescope'-example, but because of its frequency it becomes even more

[15]The percentage of triplets that were identical and correct in both the annotation sets

problematic in parsing patent language. There was high agreement on triplets containing the ATTR and OBJ relations (94% and 83% respectively). Differences in annotation were resolved by discussion, and the resulting set of annotations was used to evaluate the parser.

Parsing accuracy was rather low: On the test set of 14 sentences the parser achieved 0.37 in precision (the number of correctly generated triplets divided by the number of all generated triplets) and 0.31 in recall (the number of correctly generated triplets divided by the number of all correct triplets), so accuracy[16] rated 0.34. Thus the parser generated a lot of incorrect triplets, while it also generated too few triplets (333 generated versus 416 manually annotated). The latter is a consequence of gaps in the lexical and syntactic coverage of this version of the parser. We tested the lexical coverage and 98% of the words in the corpus featured in the parser lexicon[17] or were robustly recognized (see infra). The only words that the parser could not recognize were chemical formulae. It is difficult to say something about the syntactic coverage of the parser for this type of language: We previously tested the same parser on a general language regression test set of about 300 short sentences. On this set the parser achieved 0.87 accuracy. It is unlikely that the grammatical constructions used in patent texts are so different from those used in general language that the parser would perform so badly on this kind of text. More likely, the low accuracy is a consequence of some gaps in the grammatical coverage and the difference in language use that we observe in patent texts.

Looking at the faulty triplets, we noticed that quite often these were caused by lexical ambiguity or incomplete POS information in the lexicon: when a word is taken to be a verb, while it is in fact a noun or an adjective, this will have a profound effect on all the triplets in which a word connects with this verb. For example, during the analysis of the parser triplets, we noticed that the quality of the triplets containing SUBJ or OBJ relators was exceptionally bad (0.42 and 0.22 accuracy respectively). Analysis of the sentences showed that the erroneous interpretation of 'said' (as in 'the second screw in said device') as a verb instead of an adjective created at least four faulty triplets per occurrence, e.g. `[screw,SUBJ,said]`, `[said,OBJ,device]`, `[said,PREPin,second]`, ...

In order to be able to deal with all sorts of text, our parser is equipped with a few robust rules, which can robustly recognise words that are not in the lexicon and give them a part of speech (for example any word ending in -ly that can not be found in the lexicon, will be recognised as an adverb), or assign a part of speech to a word that is different from what is mentioned in the lexicon (for example, the fact that the verb 'run' can become a noun in 'The first run of the cycle went fine.' is covered in the grammar rules). On the one hand, such robust rules improve the recall of the parser as some of the terminology in the patent texts is not included in the lexicon and must therefore be recognised by other means. Furthermore, the language use in patents is quite different from general language use: The different POS possibilities of the word 'said' is a clear example of that. On the other hand, such robust rules must be used with caution: If used too liberally they can pose a risk for precision, because they make the parser more likely to generate faulty triplets. If any noun were allowed to be a verb and any adjective a noun or a verb, even a simple phrase like 'a good book shop' would have at least four parses with the following interpretations: 'a good shop for books', 'a good book that shops',' goods that shop for books', 'goods that book a shop'. These would render different triplets and without any extra information it would be impossible for the parser to identify the correct parse.

At this moment we are experimenting with a hybrid version of the parser in which the parsing process is guided by frequency information of good[18] dependency triplets in patent texts. This way the robustness of the parser remains intact, but the proliferation of faulty triplets is kept to a minimum.

5. DISCUSSION

In this section some analysis is done trying to identify the reasons behind the bad reranking performance of our second step in the patent retrieval task. There are three reasons why the MAP score is so much lower in the reranking experiment compared to the baseline.

First of all, it seems that Dependency Triplets – in their current form – are too detailed to be used as index terms. In the reranking experiment an average of 90.1% documents was returned. This means that for almost 10% of the documents there was no overlap between the triplets in the patent topic and the documents returned in the first retrieval step. While this specificity is problematic for the retrieval results, it is also the greatest strength of the linguistically-based system. We should find the correct balance between detailed information and more general index terms by adding extensive lexical normalisation to our system. If the triplets contain lemmas instead of word forms, a great deal of the morphological variation will disappear and overlap should increase. As we use a parser with an extensive lexicon to generate the dependency triplets, lemmatisation is not a very difficult step to implement. Another strategy would be to stem all the word forms in the dependency triplets before they are used in the retrieval process. This would be less effective than using lemmatisation: lemmatisation is more selective than stemming since a single stem can be the basis of more than one lemma; Furthermore, cropping the word forms to their stems would make the dependency triplets less informative when they are used to guide the parsing process of the hybrid parser.

A second reason why this experiment yielded negative results is the gaps in the triplet coverage of the documents. As mentioned above, the parser was not able to parse all

[16]This is the F1-measure, calculated from the precision and recall achieved by the parser.

[17]The fact that a word is found in the lexicon (lexical coverage) does not necessarily mean that the lexical information is complete and accurate for all uses and contexts. For example, if the word 'chair' is known in the lexicon only as a noun in the sentence 'He needs people to chair the first session.' where 'chair' is a verb, the parser will fail to produce the correct parse.

[18]This means reliable triplets, irrespective of the context in which they were found.

the documents completely: two topic documents completely failed to parse and in about 2% percent of the retrieved documents, the parser failed on part of the text, thus creating holes in the triplet coverage of that document, which may be crucial to the retrieval process. It is clear that we need to add another pre-processing step to our system to make sure that unparsable (large) sections are split up into smaller units that the parser can manage.[19] We estimate that the parser should have generated around 40 million triplets, instead of the 32 million that have been produced in this experiment. Triplet coverage should also improve when the grammatical coverage improves, more specifically for those structures that are typical for patent texts.

The final and probably most important reason for these low scores is the bad quality of the generated triplets. As our system depends on exact matching of detailed (and consequently low frequency) terms, lowering the frequencies with which the terms (triplets) occur by assigning some occurrences to faulty triplets has a very harmful effect on the retrieval process.

As mentioned above the language used in the claims section is very difficult to parse, even for humans, and it is quite possible that using the language from the abstract or description fields would have yielded better results for this experiment. The claims section is, however, a very important part of the patent text and our parser must be able to parse the language correctly. We are now working on a hybrid parser that uses information about triplet frequency to guide the parsing process. By supplying it with a set of correct triplets that are typical for the language used in claims, the parser should be able to deal with lexical ambiguities. Better syntactic coverage will also improve the parser's performance.

6. CONCLUSIONS

In this paper, we described a reranking experiment following our participation in the CLEF-IP 2009 track. We explored whether using syntactic structures represented by means of dependency triplets as index terms would lead to improvements in the reranking of the relevant documents that were found in the baseline run for the CLEF-IP track. Our experiment illustrated the difficulties of generating good quality triplets for retrieval purposes. We were not able to improve the ranking in the second step. On the contrary, the MAP scores were significantly lower for the reranking experiment. This was caused by the following factors: a) the overall quality of the triplets was low; b) there were gaps in the triplet coverage of the documents due to parse failures; c) there was not enough overlap between topic and corpus triplets because the triplets are too detailed in their current form.

For future work, we need to improve the parser accuracy both for lexical and syntactic ambiguity. We believe that using a hybrid parser with triplet frequency information will have a significant effect on the quality of the generated triplets. We also need to use lemmas instead of word forms in our dependency triplets in order to improve overlap between the topic and corpus documents.

When these improvements have been implemented in the parser, this experiment should be repeated in order to find conclusive evidence whether or not dependency triplets can improve the reranking of relevant documents found by a BOW approach in patent retrieval. If the results are equally poor, we will have to revisit our arguments that predict that triplets are conducive to this task.

[19]For example: the entire claims section consists of one, immense sentence. By splitting this sentence up into smaller, more manageable clauses, we could improve parsing speed and produce more triplets for this section.

7. ACKNOWLEDGMENTS

The TM4IP project is being funded by Matrixware.

8. REFERENCES

[1] M. A. Alonso, J. V. Ferro, and V. M. Darriba. On the Usefulness of Extracting Syntactic Dependencies for Text Indexing. In *AICS '02: Proceedings of the 13th Irish International Conference on Artificial Intelligence and Cognitive Science*, pages 3–11, London, UK, 2002. Springer-Verlag.

[2] A. Arampatzis, T. van der Weide, C. Koster, and P. van Bommel. *Encyclopedia of Library and Information Science*, chapter Linguistically-motivated information retrieval. Marcel Dekker, New York, 2000.

[3] Atsushi Fujii, Masao Utiyama, Mikio Yamamoto and Takehito Utsuro. Producing a Test Collection for Patent Machine Translation in the Seventh NTCIR Workshop. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA).

[4] H. BaoQuoc, T. B. T. Dong, J. Chevallet, and M. Bruandet. A structured indexing model based on noun phrases. In *RIVF*, pages 81–89, 2006.

[5] D. Bonino, A. Ciaramella, and F. Corno. Review of the state-of-the-art in patent information and forthcoming evolutions in intelligent patent informatics. *World Patent Information*, June 2009.

[6] T. Brants. Natural Language Processing in Information Retrieval. In *Proceedings of the 14th Meeting of Computational Linguistics in the Netherlands 2003*, 2004.

[7] L. Chen, N. Tokuda, and H. Adachi. A patent document retrieval system addressing both semantic and syntactic properties. In *Proceedings of the ACL-2003 workshop on Patent corpus processing*, pages 1–6, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[8] H. Cui, R. Sun, K. Li, M.-Y. Kan, and T.-s. Chua. Question answering passage retrieval using dependency relations. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and Development in Information Retrieval*, pages 400–407, 2005.

[9] E. Escorsa, M. Giereth, Y. Kompatsiaris, S. Papadopoulos, E. Pianta, G. Piella, I. Puhlmann, G. Rao, M. Rotard, P. Schoester, L. Serafini, and V. Zervaki. Towards content-oriented patent document processing. *World Patent Information*, 30(1):21–33, 2008.

[10] J. Fagan. *Experiments in Automatic Phrase Indexing*

For Document Retrieval: A comparison of Syntactic and Non-Syntactic Methods. PhD thesis, Cornell University, 1987.

[11] J. Gao, J.-Y. Nie, G. Wu, and G. Cao. Dependence language model for information retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 170–177, New York, NY, USA, 2004. ACM.

[12] E. Graf and L. Azzopardi. A methodology for building a test collection for prior art search. In *Proceedings of the 2nd International Workshop on Evaluating Information Access (EVIA)*, pages 60–71.

[13] C. S.-G. Khoo. The Use of Relation Matching in Information Retrieval. *LIBRES: Library and Information Science Research Electronic Journal*, 7(2), 1997.

[14] C. Koster and J. Beney. Phrase-Based Document Categorization Revisited. 2009.

[15] C. Koster, N. Oostdijk, S. Verberne, and E. Dhondt. Challenges in Professional Search with PHASAR. 2009.

[16] W. Kraaij and R. Pohlmann. Comparing the Effect of Syntactic vs. Statistical Phrase Indexing Strategies for Dutch. In *ECDL '98: Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries*, pages 605–617, London, UK, 1998. Springer-Verlag.

[17] H. Mase, T. Matsubayashi, Y. Ogawa, M. Iwayama, and T. Oshio. Proposal of two-stage patent retrieval method considering the claim structure. In *ACM Transactions on Asian Language Information Processing (TALIP)*, volume 4, pages 190–206, 2005.

[18] D. Metzler and W. B. Croft. A Markov random field model for term dependencies. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 472–479, New York, NY, USA, 2005. ACM.

[19] M.-F. Moens. *Automatic Indexing and Abstracting of Document Texts*, volume Vol.6 of *The Kluwer International Series on Information Retrieval*. 2005.

[20] G. Roda, J. Tait, F. Piroi, and V. Zenz. CLEF-IP 2009: Retrieval experiments in the Intellectual Property domain. 2009.

[21] S. Sheremetyeva. Towards Designing Natural Language Interfaces. In *Proceedings of the 4th International Conference "Computational Linguistics and Intelligent Text Processing"*, 2003.

[22] A. Shinmori, M. Okumura, Y. Marukawa, and M. Iwayama. Patent claim processing for readability: structure analysis and term explanation. In *Proceedings of the ACL-2003 workshop on Patent corpus processing*, pages 56–65, 2003.

[23] A. F. Smeaton and C. J. van Rijsbergen. Experiments on incorporating syntactic processing of user queries into a document retrieval strategy. In *SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 31–51, New York, NY, USA, 1988. ACM.

[24] F. Song and W. B. Croft. A general language model for information retrieval. In *CIKM '99: Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321, New York, NY, USA, 1999. ACM.

[25] Y.-I. Song, K.-S. Han, S.-B. Kim, S.-Y. Park, and H.-C. Rim. A novel retrieval approach reflecting variability of syntactic phrase representation. *Journal of Intelligent Information Systems*, 31(3):265–286, 2008.

[26] T. Strzalkowski, J. Carballo, and M. Marinescu. Natural language information retrieval: TREC-3 report. Technical report, In The Third Text Retrieval Conference (TREC 3), 1994.

[27] Y. Tseng and Y. Wu. A study of search tactics for patentability search: a case study on patent engineers. In *Proceeding of the 1st ACM workshop on Patent information retrieval*, pages 33–36, 2008.

[28] S. Verberne, L. Boves, N. Oostdijk, and P.-A. Coppen. What is not in the Bag of Words for Why-QA? 2009. To appear in *Computational Linguistics*.

[29] C. Zhai. Fast statistical Parsing of Noun Phrases for document Indexing. In *Proceedings of the fifth conference on Applied natural language processing*, pages 312–319, Morristown, NJ, USA, 1997. Association for Computational Linguistics.