Animacy Detection in Stories

Folgert Karsdorp[1], Marten van der Meulen[1], Theo Meder[1], and Antal van den Bosch[2]

1 **Meertens Institute**
Amsterdam, The Netherlands
{folgert.karsdorp,marten.van.der.meulen,theo.meder}@meertens.knaw.nl
2 **Radboud University**
Nijmegen, The Netherlands
a.vandenbosch@let.ru.nl

Abstract

This paper presents a linguistically uninformed computational model for animacy classification. The model makes use of word n-grams in combination with lower dimensional word embedding representations that are learned from a web-scale corpus. We compare the model to a number of linguistically informed models that use features such as dependency tags and show competitive results. We apply our animacy classifier to a large collection of Dutch folktales to obtain a list of all characters in the stories. We then draw a semantic map of all automatically extracted characters which provides a unique entrance point to the collection.

1998 ACM Subject Classification I.2.7 Natural Language Processing

Keywords and phrases animacy detection, word embeddings, folktales

Digital Object Identifier 10.4230/OASIcs.CMN.2015.82

1 Introduction

For almost all species in the world, the capacity to distinguish animate objects from inanimate objects is essential to their survival. Those objects could be prey, for example, or predators, or mates. The fundamental nature that the distinction between animate and inanimate has for humans is reflected in the fact that this division is acquired very early in life: children of less than six months old are well able to distinguish the two categories from one another [16]. Moreover, recent brain research shows that the distinction appears in the organization of the brain (e.g. [8]). For some researchers, this provides evidence for the idea that the division between animate and inanimate is an innate part of how we see the world.

Although animacy may be a scalar rather than a strictly categorical distinction (see e.g. the animacy hierarchy in [4] and research such as [25]), the animate/inanimate distinction is traditionally taken as binary with regard to lexical items: something is either animate (e.g. a human) or not (e.g. a chair). This standpoint has been challenged, however, by researchers from different fields. Firstly, it has long been established in linguistic typology that not all languages award animacy to the same entities in different grammatical categories. As [4] notes, many languages, such as, for example, English, distinguish between human and not-human in the choice of pronouns; other languages, such as Russian, distinguish between animate (entailing humans and animals) versus non-animate (entailing everything else) in their interrogative pronouns. This indicates different subdivisions of animacy in the respective languages. Secondly, philosophers such as Daniel Dennett support the view that animacy and aliveness are to be treated as epistemological stances rather than fixed states in the world: not ineffable qualia but behavioral capacity defines our stance towards objects [6].

6th Workshop on Computational Models of Narrative (CMN'15).
Editors: Mark A. Finlayson, Ben Miller, Antonio Lieto, and Remi Ronfard; pp. 82–97

OpenAccess Series in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In other words, depending on whether people *think* that an object is animate, they utilize different cognitive strategies to explain and predict the actions of those objects. Finally, evidence from psycholinguistic research has accumulated to support this view of animacy as a cognitive viewpoint rather than an extra-perceptive absolute. Nieuwland & Berkum [15] for example show that college student test subjects readily accept animate behavior from inanimate objects within the proper contexts, and Vogels et al. [9] moreover emphasize the relation between animacy and motion, showing that factors such as self-propelment play a crucial role in recognizing or awarding animacy to certain objects. This is exemplified in the opening of this well-known story:[1]

> *A farmer bought a pancake on the market. Once he got home, the farmer was hungry and began to bake the pancake. The farmer tried one of his skillful flipping techniques, but he failed and the pancake fell on the ground. Coincidentally, the door of the kitchen was open and the pancake rolled out to the field, as hard as he could...*

Although initially, based on their knowledge of the world, readers will regard the pancake as inanimate, the self-propelled motion verb 'rolled' initiates our shift towards an animate interpretation of the pancake. As readers (or listeners) of a story, we choose to view participating objects at varying levels of abstraction in order to predict their behavior. Dennett [6] defines three levels of abstraction: (1) the *physical stance*, (2) the *design stance* and (3) the *intentional stance*. The physical stance deals with predictions about objects given their physical properties. The design stance deals with concepts such as purpose, function or design. The intentional stance is concerned with belief, thinking and intentions. These are all cognitive strategies we use to predict and explain the actions of objects in our environment. Interestingly, in the process of reading the opening of the story about the fleeing pancake, readers and listeners experience the transition from one strategy to the next quite clearly. Initially, the pancake is interpreted from a physical stance, or perhaps the more abstract design stance in terms of the purpose (i.e. to stave off hunger). It is only at the last adverbial phrase 'as hard as he could' that we start to wonder whether we should adopt to the yet more abstract intentional stance and consider the pancake to be a rational agent.

Given the fundamental nature of the distinction between animate and inanimate, it is perhaps not too surprising that it has proven to be useful in a variety of natural language processing tasks dealing with e.g. anaphora resolution and dependency parsing [18, 11, 22]. Existing methods for the automatic labeling of text for animacy are usually rule-based, machine-learning-based, or a hybrid of these methods. Common to most approaches is the fact that they make use of semantic lexicons with information about animacy, as well as syntactic cues in a text. Both feature types are relatively costly to obtain as they require lexical resources or syntactic parsing systems, which, with the exception of a few languages, are not readily available.

In this paper we present a new linguistically uninformed model to automatically label texts for animacy. We show that we can do away with features that require syntactic parsing or semantic lexicons while still yielding competitive performance. We focus on labeling animacy in stories because stories pose some particularly interesting problems to automatic systems of animacy recognition. As the example of the fleeing pancake already illustrated, in stories any entity may at some point exhibit animate behavior, even when they are inanimate in the 'real' world. Another example is the *Sorcerer's Apprentice* sequence in Walt Disney's

[1] `http://www.verhalenbank.nl/items/show/9636`

famous *Fantasia*, in which brooms display the ability to collect buckets of water. Such examples, where pancakes, brooms and other entities act as animate beings, make a clear case for developing dynamic, data driven systems that do not rely too much on static and fixed world knowledge, but rather on immediate context.

The remainder of this paper is structured as follows. We will start with a short overview of existing techniques for automatically labeling animacy in texts, including the definitions of animacy used in these papers (§2). After a description of the corpus used in our study and how the annotations of the corpus have been established (§3), we will give an account of our computational models in Section 4. We report on the empirical results in Section 5. Next, we provide an evaluation on a larger dataset, while also showing a real-world application of our animacy detection system (§6). The final section offers our conclusions and possible directions for future research.

2 Previous Work

A handful of papers deal with automatic animacy detection. Most approaches make use of rule-based systems or machine learning systems with morphological and syntactic features. [7] present a rule-based system that makes use of the lexical-semantic database WordNet. They label each synset in WordNet for animacy. Using a variety of rules to detect the head of an NP, they use the fraction of synsets in which a particular noun occurs to arrive at a classification for animacy. [17] extend their previous algorithm by first determining the animacy of senses from WordNet on the basis of an annotated corpus. They then apply a k-nearest neighbor classifier using a number of lexical and syntactic features alongside features derived from WordNet to arrive at a final animacy classification.

[19, 20, 21] present a number of animacy classifiers that make use of syntactic and morphological features. These features include the frequency of analysis of the noun as 'subject' or 'object', the frequency of the occurrence of a noun in a passive *by*-phrase, and the frequency of the noun as a subject followed by either animate personal pronouns or inanimate personal pronouns. These features are then aggregated for each lemma after which a machine learning system (decision tree or k-nearest neighbor classifier) is trained. A similar approach is presented in [3]. In this study a Maximum Entropy classifier is trained on the basis of three feature types: (1) bag-of-words with and without their corresponding Part-of-Speech tags, (2) internal syntactic features such as the syntactic head and (3) external syntactic features that describe the dependency relation of a noun to a verb (i.e. subject relation, object relation etc.) This is the only study that makes use of a corpus fully labeled for animacy. In an approach partially related to animacy detection, [10] attempt to extract the cast (i.e. all characters) from a story. Similar to [3] they rely on dependency tags to extract the subjects of direct and indirect speech.

[1] present a model that attempts to generalize the animacy information in a lexical-semantic database of Dutch by augmenting 'non-ambiguous' animate entries with contextual information from a large treebank of Dutch. They apply a k-nearest neighbor algorithm with distributional lexical features that aim to capture the association between a verb or adjective and a particular noun. The idea is that nouns that occur in similar contexts as animate nouns are more likely to be animate than nouns that occur more frequently in contexts similar to inanimate nouns.

[14] present an approach that combines a number of animacy classifiers in a voting scheme and aims at an interpretable and correctable model of animacy classification. A variety of classifiers is used, such as the WordNet-based approach of [7], named entity recognition

systems, and dictionary sources.

The approaches mentioned above present us with a number of problems. First, nearly all of them rely heavily on costly, linguistically informed features derived from lexical-semantic databases or syntactic parsing. For most languages in the world, however, we cannot rely on these resources, either because they do not exist, or because their performance is insufficient. Second, animacy detection is often seen as a useful feature for a range of natural language processing techniques, such as anaphora resolution and syntactic parsing. The mutual dependence between these techniques and animacy detection, however, is in fact a chicken-and-egg situation.

Another major problem with the approaches above is, as said earlier, that they are lemma-based, which means that the models are generally insensitive to different usages of a word in particular contexts. In other words, in most of the literature on automatic animacy detection, a static, binary distinction is made between animate and inanimate. [3] for example, define objects as animate if they are alive and have the ability to move under their own will. [18] define animacy in the context of anaphora resolution: something is animate "if its referent can also be referred to using one of the pronouns he, she, him, her, his, hers, himself, herself, or a combination of such pronouns (e.g. his/her)". However, as was explained above, these definitions are not necessarily in line with current linguistic and neurological research [15]. Similarly, they are not particularly applicable to the rich and wondrous entities that live in the realm of stories. As was shown above, although a pancake is typically not an animate entity, its animacy depends on the story in which it appears, and even within the story the animacy may change. To accommodate this possibility, we therefore choose to define animacy in terms of Dennett's intentional stance, which is more dynamic, and which ultimately comes down to the question whether "you decide to treat the object whose behavior is to be predicted as a rational agent" [6, pp. 17]. Our system for animacy detection therefore needs to be dynamic, data driven, and token-based. It may to some extent rely, but cannot rely too heavily, on static world knowledge.

3 Data, Annotation and Preprocessing

To develop this dynamic data-driven system we use a corpus of Dutch folktales. As argued in the introduction, our reason to use folktales is that, as [9] note, 'In cartoons or fairy tales [...] inanimate entities or animals are often anthropomorphized', which means that the material could yield interesting cases of unexpected animacy, as is the case with the pancake in *The fleeing pancake* and the broomsticks in *Fantasia*.

Our initial corpus consists of 74 Dutch stories from the collection *Volkssprookjes uit Nederland en Vlaanderen*, compiled by [27]. The collection is composed of Dutch and Flemish retellings of popular and widespread stories, including such tales as *The Bremen Town Musicians* (ATU 130) [2] and *The Table, the Ass, and the Stick* (ATU 563), as well as lesser-known stories such as *The Singing Bone* (ATU 780) and *Cock, Hen, Duck, Pin, and Needle on a Journey* (ATU 210). This last story is again a clear example where otherwise inanimate objects are animated, as it concerns the adventures of several household items, such as a *pin*, a *hackle*, an *egg*, and a *whetstone*. A digital version of the collection is available in the Dutch Folktale Database from the Meertens Institute (corpus SINVSUNV.20E).[3]

[2] The ATU numbers refer to the classificatory system for folklore tales, as designed by Aarne, Uther and Thompson [28].

[3] See `http://www.verhalenbank.nl`

Using a single collection for our corpus presents us with a helpful homogeneity with regard to the editor, length of the stories, and language use, as well as exhibiting some content-wise diversity among the collection, which contains fairytales and legends.

All together, the corpus consists of 74,504 words, from 5,549 unique words. Using the annotation tool brat (brat rapid annotation tool), an online environment for collaborative editing[4], two annotators labeled words for animacy, within the context of the story.[5] All unlabeled words were implicitly considered to be inanimate. The following sentence provides an example annotation.

(1) Jij smid, jij bent de sterkste; hou je vast aan de bovenste
ANIMATE ANIMATE ANIMATE ANIMATE
takken, en dan ga jij kleermaker aan zijn benen hangen en zo gaan
ANIMATE ANIMATE ANIMATE
we maar door
ANIMATE
'You, blacksmith, you are the strongest; hold on to the upper branches and then, you, tailor, will grab his legs and so we go on…'

Because we interpreted animacy within the context of the story, the same lexical item could be labeled differently in different stories. For example, in the above-mentioned example of the pancake, which occurs in SINVS076 in our corpus, the pancake is tagged consistently as 'animate'. In another story, SINVS042, where at one point a soldier is baking pancakes, the pancakes do not act, and are thus not labeled as 'animate'. The following sentences show how this was employed in practice.

(2) Terwijl hij de pannekoek bakte, keek hij naar het ding, dat uit de
ANIMATE ANIMATE
schouw gevallen was

'While he was baking the pancake, he looked at the thing, which had fallen from the hearth…'

(3) Toevallig stond de deur van de keuken open en de pannekoek rolde naar buiten,
ANIMATE
het veld in, zo hard hij maar kon.
ANIMATE
'Coincidentally the door of the kitchen was open and the pancake rolled outside, into the field, as fast as it could'

This annotation resulted in 11,542 animate tokens of 743 word types, while implicitly yielding 62,926 inanimate tokens from 5,011 unique inanimate words. Because of our context-dependent approach, some words, such as *pancake* and *egg*, occurred in both animate types as inanimate types, because they were labeled as both animate and inanimate in some cases in our corpus. It is telling that of the animate tokens 4,627 (40%) were nouns and proper nouns, while only 6,878 of the inanimate tokens (11%) are nouns. This shows that being a noun is already somewhat of an indication for animacy. After tokenization with the tokenization

[4] `http://brat.nlplab.org`
[5] On the basis of five stories that were annotated by both annotators we computed an inter-annotator agreement score (Cohen's Kappa) of $K = 0.95$.

module of the Python software package Pattern [5] we fed all stories to the state of the art syntactic parser for Dutch, Alpino [2]. From the resulting syntactic parses, we extracted the features for the linguistically informed models, see Section 4.3.

4 Experimental Setup

This section describes our experimental setup including the features used, the machine learning models we applied, and our methods of evaluation.[6].

4.1 Task description

We formulate the problem of animacy detection as a classification problem where the goal is to assign a label at word level, rather than at lemma level. This label indicates whether the word is classified as animate or inanimate.

4.2 Evaluation

Inanimate words far outnumber animate words in our collection (see §3). Reporting accuracy scores would therefore provide skewed results, favoring the majority category. The relative rarity of animate words makes evaluation measures such as the well-known $F1$-score more appropriate. For this reason, we report on the precision, recall and $F1$-score [30] of both classes for all experiments. Also, while in most of the literature on animacy detection results are only presented for the classification of nouns or noun phrases, we will, while reporting on nouns and noun phrases as well, additionally report on the results for all words in a text.

In real-world applications an animacy detection system will most likely be faced with completely new texts instead of single words. It is therefore important to construct a training and test procedure in such a way that it mimics this situation as closely as possible. If we would, for example, make a random split of 80% of the data for training and 20% for testing on the word level, we run the risk of mixing training data with test data, thereby making it too easy for a system to rely on words it has seen from the same text. [3] fall into this trap by making a random split in their data on the sentence level. In such a setup, it is highly likely that sentences from the same document are present in both the training data and the test data, making their evaluation unrealistic. To circumvent this problem, we split the data at the story level. We make use of 10-fold cross-validation. We shuffle all stories, partition them in ten portions of equal size. In ten iterations, each partition acts as a test set, and the other nine partitions are concatenated to form the training set.

4.3 Features

We explore a range of different features and feature combinations including lexical features, morphological features, syntactic features, and semantic features.

4.3.1 Lexical features

We take a sliding-window approach where for each focus word (i.e. the word for which we want to predict whether it is animate or not) we extract both n words to the left and n

[6] The data set and the code to perform the experiments are available from `https://fbkarsdorp.github.io/animacy-detection`

words to the right, as well as the focus word itself. In all experiments we set n to 3. In addition to the word forms, for each word in a window we also extract its lemma as provided by the output of the syntactic parser Alpino.

4.3.2 Morphological Features

For each word we extract its part-of-speech tag. For reasons of comparability we choose to use the tags as provided by Alpino, instead of a more specialized part-of-speech tagger. Again, we take a sliding window approach and extract the part-of-speech tags for three words left and right of the focus word, as well as the tag of the focus word itself.

4.3.3 Syntactic Features

We extract the dependency tag for each word and its $n = 3$ neighbors to the right and to the left as provided by the syntactic parser Alpino. Animate entities tend to take the position of subject or object in a sentence which is why this feature is expected and has proven to perform rather well.

4.3.4 Semantic Features

The most innovative feature we have included in our model is concerned with semantic similarity. In his *Philophische Untersuchungen* Wittgenstein already suggests that "Die Bedeutung eines Wortes ist sein Gebrauch in der Sprache"[7] (PI 43). This is reflected by the well-known insight in computational linguistics that the meaning of words can be approximated by comparing the linguistic contexts in which words appear. In other words: words that often co-appear with the same set of words, will have a more similar meaning. Recently, there has been a lot of interest in procedures that can automatically induce so-called 'word embeddings' from large, unannotated collections of texts (e.g. [13, 24]). These models typically attempt to learn vector representation with less dimensions than the vocabulary size for each word in the vocabulary which captures the typical co-occurrence patterns of a word in the corpus. The similarity between words can then be approximated by applying similarity metrics, such as the cosine metric, to these vectors of word embeddings.

We have trained word embeddings with 300 dimensions using the popular skip-gram architecture [13] on the Dutch corpus of COW (COrpora from the Web). COW is a collection of linguistically processed web corpora for English, Dutch, Spanish, French, Swedish and German [26]. The 2014 Dutch corpus contains 6.8 billion word tokens. The idea behind using the word embeddings is that similarities between animate words can be estimated by inspecting the context in which they occur. From this follows, for example, that the word embeddings of an animate word are more similar to those of other animate words, as opposed to the embeddings of inanimate words.

To give an illustration of this idea, in Figure 1 we depict a two-dimensional Principle Component Analysis (PCA) projection of the 300 dimensional word embedding vectors for a number of typically animate and typically inanimate words. The horizontal gray line in the plot illustrates the separability of the animate and inanimate words in the first dimension of the PCA projection. It is interesting to observe that *ghost* is the one closest to all other inanimate entities. Likewise, words such as *castle*, *house* or *car* are often used in figurative language (metonymy), for example to refer to the people owning or living in the

[7] The meaning of a word is its use in the language.

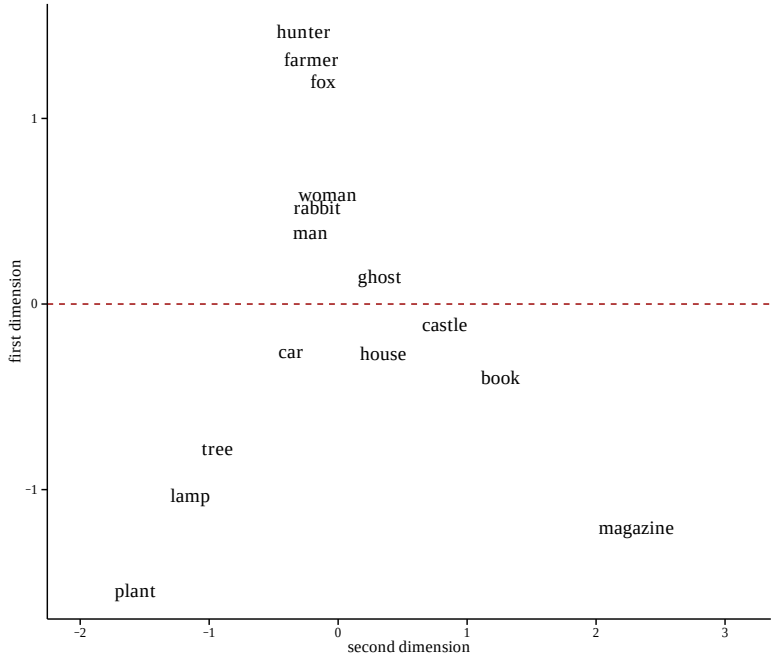

Figure 1 Two-dimensional PCA projection of the 300 dimensional word embedding vectors for a number of animate and inanimate words. The horizontal line illustrates the separability between the two classes in the first dimension.

castle. Perhaps this ambiguous animacy position is responsible for their position in the first dimension close to real animate entities.

4.4 Models

We employ a Maximum Entropy classifier with L2 regularization as implemented in [23]. In all experiments, we set the regularization strength parameter C to 1.

We compare nine models in which we make use of different feature combinations: (1) words, (2) words and Part-of-Speech tags, (3) words, Part-of-Speech tags and lemmata, (4) words, Part-of-Speech tags, lemmata and dependency tags, (5) word embeddings and (6-9) the features in model 1 to 4 with word embeddings.

Although our background corpus is sufficiently large to cover most words in an unseen text, there will always be rare words for which we do not have learned word embeddings. Therefore, in order to effectively make use of the word embedding vectors, we need a way to deal with out-of-vocabulary items. We adopt a simple strategy where we make use of a primary classifier and a back-off classifier. For models 6 to 9, we augment each word with its corresponding 300 dimension word embeddings vector. In the case of out-of-vocabulary words, we resort to a back-off model that contains all features except the word embeddings. For example, a model that makes use of words and word embeddings, will make a prediction on the basis of the word features alone. In case of the model that solely uses the embeddings (model 5), the back-off classifier is a majority-vote classifier, which classifies unseen words as inanimate.

Table 1 Precision, Recall and $F1$-score for animate and inanimate classes per feature setting for all words.

	inanimate			animate		
	P	R	$F1$	P	R	$F1$
embeddings	0.98	0.99	0.98	0.93	0.89	0.91
word	0.96	0.99	0.98	0.94	0.78	0.85
word + embeddings	0.98	0.99	0.98	0.94	0.90	0.91
word + PoS	0.97	0.99	0.98	0.94	0.86	0.89
word + PoS + embeddings	0.98	0.99	0.99	0.94	0.91	0.93
word + PoS + lemma	0.97	0.99	0.98	0.94	0.86	0.90
word + PoS + lemma + embeddings	0.98	0.99	0.99	0.94	0.91	0.93
word + PoS + lemma + dep	0.97	0.99	0.98	0.94	0.86	0.90
word + PoS + lemma + dep + embeddings	0.98	0.99	0.99	0.94	0.92	0.93

5 Results

In Table 1 we present the results for all nine models on the complete data set. For each model we report the precision, recall and $F1$-score for the animate words and the inanimate words.

All models perform well on classifying inanimate words. However, since this is the majority class, it is more interesting to compare the performance of the models on the animate instances. It is interesting to observe that the 'simple' n-gram word model already performs rather well. Adding more features, such as Part-of-Speech or lemmata, has a consistently positive impact on the recall of the model, while leaving the precision untouched. As can be observed from the table, employing the rather expensive dependency features shows barely any improvement.

The model that only uses word embedding features is one of the best performing models. This is a context-insensitive model that operates on the level of the vocabulary, which means that it will predict the same outcome for each token of a particular word type. The high precision and high recall show us that this model has acquired knowledge about which words *typically* group with animate words and which with inanimate words. However, the models that combine the word embeddings with the context sensitive features, such as word n-grams or Part-of-Speech tags, attain higher levels of precision than the context-insensitive model. The best performance is achieved by the model that combines the word features, Part-of-Speech tags and the word embeddings. This model has an $F1$-score of 0.93 on animate words and 0.99 on inanimate words. Adding more features does not result in any more performance gain.

Table 2 zooms in on how well nouns and names are classified. The best performance is again achieved by the model that combines the word features with the part-of-speech tags and word embeddings, resulting in an $F1$-score of 0.92 for animate instances and 0.95 for inanimate instances. The relatively lower score for the inanimate class can be explained by the fact that relatively easy instances, such as function words, which are never animate, are not included in the score now.

Table 2 Precision, Recall, and $F1$ score for animate and inanimate classes per feature settings for all words tagged as noun.

	inanimate			animate		
	P	R	$F1$	P	R	$F1$
embeddings	0.90	0.96	0.92	0.93	0.85	0.89
word	0.78	0.98	0.87	0.96	0.60	0.74
word + embeddings	0.90	0.97	0.93	0.95	0.85	0.90
word + PoS	0.86	0.96	0.90	0.93	0.78	0.84
word + PoS + embeddings	0.93	0.96	0.95	0.95	0.90	0.92
word + PoS + lemma	0.87	0.96	0.91	0.94	0.80	0.86
word + PoS + lemma + embeddings	0.93	0.96	0.94	0.95	0.89	0.92
word + PoS + lemma + dep	0.87	0.96	0.91	0.93	0.80	0.86
word + PoS + lemma + dep + embeddings	0.93	0.96	0.95	0.95	0.90	0.92

6 A Semantic Map of Animate Entities in the Dutch Folktale Database

Our approach to animacy classification appears to be successful. In this section we employ our classification system to extract all animate entities from unannotated folktales from the Dutch Folktale Database, all of which were not used in the previous experiment.[8] The reason for this is twofold. First, it allows us to further our evaluation of the classifier. In a classical evaluation setup – as with our approach – it is general practice to train a computational system on some training data. The performance of the system is then evaluated on a held-out test set. Our annotated corpus contains a reasonably diverse set of stories in terms of genre, yet it is fairly small and rather homogeneous in style. Even though we performed a cross-validation experiment, there is a chance of 'overfitting' to the style of the subset of folktales we trained on. The second reason for applying the classifier to such a large collection is to enrich the collection with a character-based information layer, allowing researchers to browse the collection in new ways.

6.1 Data

For our evaluation we make use of a sub-collection of folktales from the Dutch Folktale Database. The complete collection consists of about 42,000 folktales [12], and contains stories from various genres (e.g. fairytales, legends, urban legends, jokes, personal narratives) in standard Dutch and Frisian, as well as in a number of dialectal variants. Every entry in the database contains meta-data about the story, including language, collector, place and date of narration, keywords, names, and sub-genre. For our paper we make use of a sub-collection comprising 16,294 stories written in standard Dutch. The distribution of genres in the subcollection is the following: urban legends ($n = 2,795$), legends ($n = 299$), jokes ($n = 3,986$), personal narratives ($n = 693$), riddles ($n = 1,626$), sagas ($n = 6,045$) and fairy tales ($n = 832$). We evaluate a random sample of this sub-collection ($n = 212$) in which this genre distribution is taken into account.

[8] `http://www.verhalenbank.nl`

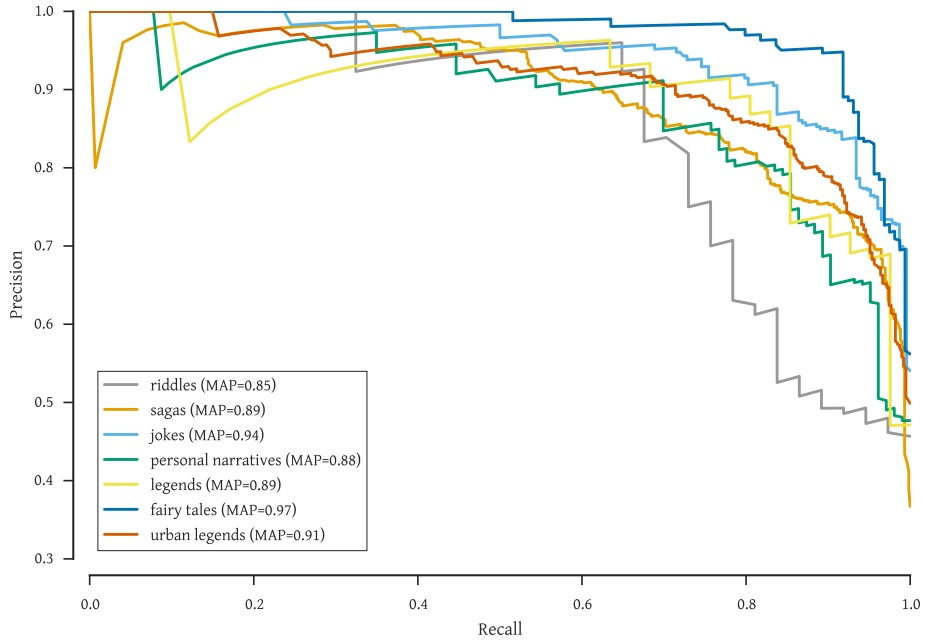

Figure 2 Precision-Recall Curves and Mean Average Precision scores per genre.

6.2 Evaluation

Our definition of animacy allows us to utilize our animacy detection system to extract all characters from a story in a similar vein as [10]. The system labels each noun and name in a text for animacy. After removing duplicate words, this produces a set of words that comprises the cast of a story. Without gold standard annotations, however, we can only evaluate these character sets for precision and not for recall. An alternative approach is to produce a ranking of all words in a story where the goal is to allocate the highest ranks to animate entities. This allows us to evaluate individual rankings using *Average Precision* which computes the average over precision scores at increasing points of recall. We compute the *Average Precision* as follows:

$$\text{AP} = \frac{\sum_{k=1}^{n}(P(k) \times rel(k))}{\text{number of relevant items}} \tag{1}$$

where k is the position in the ranked list of n retrieved items. $P(k)$ represents the precision at k and $rel(k) = 1$ if the item at k is relevant, $rel(k) = 0$ otherwise.

Per genre, a Mean Average Precision (MAP) can be computed as the normal average of the AP values of all instances within the genre.

Naturally, with this evaluation method, we still need to manually evaluate the rankings. By using a rank cutoff and evaluating a sample of all automatically annotated stories, we reduce the costly manual labor to a minimum. We order all nouns and names in a story using the output of the probabilistic decision function of the Maximum Entropy classifier. After removing duplicate words, this produces a final ranking. The rankings are evaluated with a rank cutoff at 50.

Figure 3 Visualization of characters in the Dutch Folktale Database based on their embeddings using t-SNE.

6.3 Results

We present the results in Figure 2 in which we show the Precision-Recall curve as well as the Mean Average Precision (MAP) score for each genre. The Precision-Recall curve is obtained from computing precision-recall pairs for different probability thresholds. The system performs well, especially on fairytales (MAP= 0.97) and jokes (MAP= 0.94).[9] The lowest performance is measured on riddles (MAP= 0.85). This lower score is partly due to the system's inability to position the word *blondje* ('dumb blond' with a pejorative connotation) high up the ranking.

6.4 A Semantic Map of Characters

The word embeddings that we used as features for our animacy classifier can be employed to describe the similarities and dissimilarities between the extracted animate entities. In Figure 3 we present a two-dimensional semantic map that depicts the (dis)similarities between all extracted animate entities.[10] The dimension reduction was performed using t-Distributed Stochastic Neighbor Embedding (t-SNE) [29]. The coloring of the nodes was obtained by applying a k-Means cluster analysis (k=8) to the word embeddings.

The map discloses a rich diversity of animate entities grouped into semantically coherent clusters. The pink cluster on the far left represents a grouping of all kinds of animals. Note that within this cluster there exist many subtle sub-clusters describing more specific positions in the animal taxonomy, e.g. birds and livestock, marine life, and insects. The central green cluster is occupied by characters of different professions. There is a large number of characters from the hospitality industry, such as waiter and cook, as well as from the transport sector, such as chauffeur and train conductor. One of the interesting groupings is located at the very bottom of the map. This cluster describes magical, supernatural and Christian characters (henceforth supernatural cluster). In Figure 4 we provide a detailed view of this cluster.

The supernatural cluster is noteworthy because it is, like the animal cluster, highly structured. Several clear hierarchically ordered clusters are discernible in Figure 4, with several subgroups emerging. The lower right hand corner for example entails religious or even Christian professions, such as 'bishops' and 'vicar'. From there, a link is made via 'catholics' and 'protestants' to the more general 'believers' and 'followers'. This mini-node bifurcates into two different nodes. Firstly, in the middle-right, a cluster is found containing words designating followers of different religions, such as 'Jew' and 'Muslim', which branches of to the top right node, which is a 'religious fringe' node, containing 'cult', 'satanist' and 'Freemasons'. It is interesting that 'wicca', which might be expected to be clustered in this node, as it also represents an organized semi-religious group, is clustered rather with 'magic' and 'witchcraft' in the upper-left 'magic' cluster.

The other cluster connected to the 'believers' and 'followers'-mini node is structurally complex, starting with such terms as 'people' and 'believers', but also containing, strikingly, 'Allah'. Taking into account that the Christian term 'lord' is clustered elsewhere, with adjectives such as 'compassion' and 'glory', but also with 'persecutors', this means that the two deities are embedded very differently. The cluster then continues through 'Satan'

[9] A MAP of 0.97 means that on average, nearly all actual cast members of a folktale are ranked on top, with the first case of a non-animate entity entering the ranking at about rank 5 or 6 on average.

[10] Readers are invited to view an interactive version of the map at the following address: `http://fbkarsdorp.github.io/animacy-detection/`.

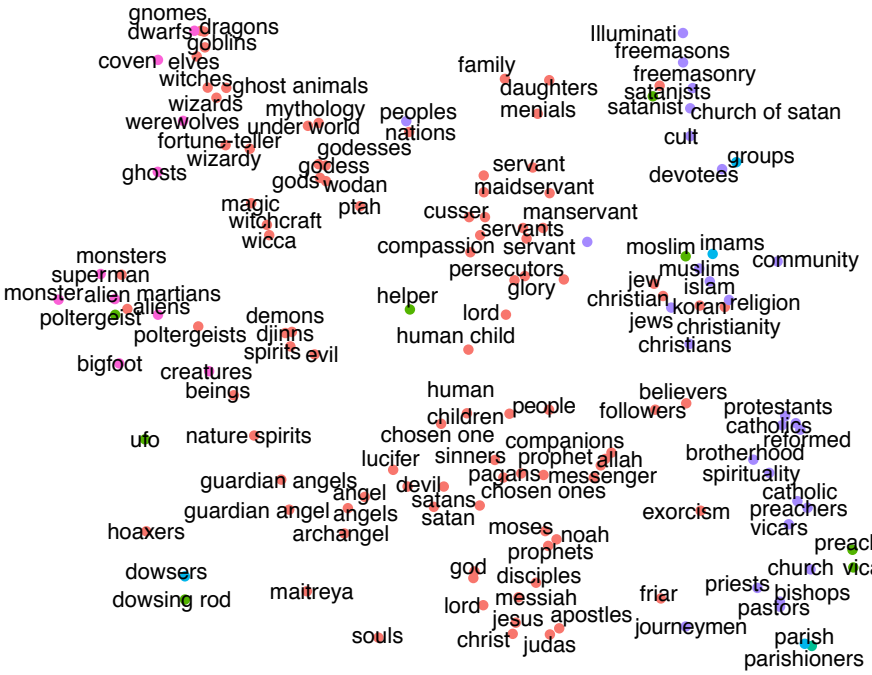

Figure 4 Detailed view of the 'Supernatural' cluster.

and 'Lucifer' to 'angels' and 'guardian angels'. These words form again a bridge towards more esoteric creatures, such as 'nature spirits', culminating in the far left 'martians' and 'superman'. This cluster is connected to the upper left hand cluster, which contains traditional magical creatures such as 'werewolves' and 'dragons'.

In summary, the semantic map makes a case for the successfulness of our approach. The word embeddings combined with the strength of t-SNE to position the characters on a two-dimensional map, yield a powerful representation. The above description is only part of the extremely rich network of associations this semantic map displays.

7 Concluding Remarks

The approach taken in this paper to create a model for animacy classification using linguistically uninformed features proves to be successful. We compared the performance of linguistically informed models (using features such as Part-of-Speech and dependency tags) to models that make use of lower-dimensional representations of the data. With the exception of the model that solely makes use of these representations, all models benefit from adding these features. The model that requires the least linguistic information (word n-grams plus word embeddings) outperforms all linguistically informed models (without embeddings). The best results are reported by the model that combines word n-grams with Part-of-Speech n-grams and word embeddings.

We have the following recommendation for future research. Natural language processing models such as co-reference resolution or linguistic parsing could benefit from a module that filters animate from inanimate candidate words. Since these models typically depend on

linguistic features, it is important that additional features, such as animacy, are not dependent on these features as well. Our linguistically uninformed model for animacy detection provides such an independent module.

The digitalization of large-scale cultural heritage collections such as the Dutch Folktale Database is often accompanied with traditional (text-based) search engines. We hope that our example of a semantic map of characters inspires researchers to disclose such collections in different and innovative ways.

Acknowledgments The work on which this paper is based has been supported by the Computational Humanities Programme of the Royal Netherlands Academy of Arts and Sciences, under the auspices of the Tunes & Tales project. For further information, see `http://ehumanities.nl`.

References

1 Jelke Bloem and Gosse Bouma. Automatic animacy classification for Dutch. *Computational Linguistics in the Netherlands Journal*, 3:82–102, 2013.

2 Gosse Bouma, Gertjan Van Noord, and Robert Malouf. Alpino: Wide-coverage computational analysis of dutch. *Language and Computers*, 37(1):45–59, 2001.

3 Samuel Bowman and Harshit Chopra. Automatic animacy classification. In *Proceedings of the NAACL - HLT 2012 Student Research Workshop*, pages 7–10, 2012.

4 Bernard Comrie. *Language Universals and Linguistic Typology*. University of Chicago Press, 2nd edition, 1989.

5 Tom De Smedt and Walter Daelemans. Pattern for Python. *Journal of Machine Learning Research*, 13:2031–2035, 2012.

6 Daniel Dennett. *The Intentional Stance*. Cambridge, Massachusetts: The MIT Press, 1996.

7 Richard Evans and Constantin Orăsan. Improving anaphore resolution by identifying animate entities in texts. In *Proceedings of the Discourse Anaphora and Reference Resolution Conference*, pages 154–162, 2000.

8 Tao Gao, Brian Scholl, and Gregory McCarthy. Dissociating the detection of intentionality from animacy in the right posterior superior temporal sulcus. *The Journal of neuroscience: the official journal of the Society for Neuroscience*, 32(41):14276–14280, 2012.

9 Emiel Krahmer Jorrig Vogels and Alfons Maes. When a stone tries to climb up a slope: the interplay between lexical and perceptual animacy in referential choices. *Frontiers in Psychology*, 4(154):1–15, 2013.

10 Folgert Karsdorp, Peter Van Kranenburg, Theo Meder, and Antal Van den Bosch. Casting a spell: Indentification and ranking of actors in folktales. In F Mambrini, M Passarotti, and C Sporleder, editors, *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities (ACRH-2)*, pages 39–50, 2012.

11 Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4), 2013.

12 Theo Meder. From a dutch folktale database towards an international folktale database. *Fabula*, 51(1–2):6–22, 2010.

13 Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *Proceedings of Workship at ICLR*, 2013.

14 Joshua Moore, Christopher Burges, Erin Renshaw, and Wen tau Yih. Animacy detection with voting models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 55–60, 2013.

15 Mante S. Nieuwland and Jos J.A. van Berkum. When Peanuts Fall in Love: N400 Evidence for the Power of Discourse. *Journal of Cognitive Neuroscience*, 18(7):1098–1111, 2005.

16 John Opfer. Identifying living and sentient kinds from dynamic information: The case of goal-directed versus aimless autonomous movement in conceptual change. *Cognition*, 86(2):97–122, 2002.

17 Constantin Orăsan and Richard Evans. Learning to identify animate references. In Walter Daelemans and Rémi Zajac, editors, *Proceedings of CoNLL-2001*, pages 129–136, Toulouse, France, July, 6 – 7 2001.

18 Constantin Orăsan and Richard Evans. Np animacy identification for anaphora resolution. *Journal of Artificial Intelligence Research*, 29:79–103, 2007.

19 Lilja Øvrelid. Animacy classification based on morphosyntactic corpus frequencies: Some experiments with Norwegian nouns. In Kiril Simov, Dimitar Kazakov, and Petya Osenova, editors, *Proceedings of the Workshop on Exploring Syntactically Annotated Corpora*, pages 24–34, 2005.

20 Lilja Øvrelid. Towards robust animacy classification using morphosyntactic distributional features. In *Proceedings of the EACL 2006 Student Research Workshop*, pages 47–54, 2006.

21 Lilja Øvrelid. Linguistic features in data-driven dependency parsing. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL 2008)*, pages 25–32, 2008.

22 Lilja Øvrelid and Joakim Nivre. When word order and part-of-speech tags are not enough – Swedish dependency parsing with rich linguistic features. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 447–451, 2007.

23 Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

24 Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of The 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar, 2014.

25 Anette Rosenbach. Animacy and grammatical variation – findings from english genitive variation. *Lingua*, 118:151–171.

26 Roland Schäfer and Felix Bildhauer. Building large corpora from the web using a new efficient tool chain. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 486–493, Istanbul, 2012. ELRA.

27 Jacques Sinninghe. *Volkssprookjes uit Nederland en Vlaanderen.* Kruseman, Den Haag, 1978.

28 Hans-Jörg Uther. *The Types of International Folktales: a Classification and Bibliography Based on the System of Antti Aarne and Stith Thompson*, volume 1-3 of *FF Communications.* Academia Scientarium Fennica, Helsinki, 2004.

29 Lauren Van der Maaten and Geoffrey Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, pages 2579–2605, 2008.

30 Cornelis Van Rijsbergen. *Information Retrieval.* Butterworths, 1979.