Computational Linguistics in the Netherlands Journal 3 (2013) 82-102 Submitted 06/2013; Published 12/2013

Automatic animacy classification for Dutch

Jelke Bloem* J.BLOEM@UVA.NL
Gosse Bouma** G.BOUMA@RUG.NL

*Dutch Linguistics, University of Amsterdam
**Information Science, University of Groningen

Abstract

We present an automatic animacy classifier for Dutch that can determine the animacy status of nouns — how alive the noun's referent is (human, inanimate, etc.). Animacy is a semantic property that has been shown to play a role in human sentence processing, felicity and grammaticality. Although animacy is not marked explicitly in Dutch, we expect knowledge about animacy to be helpful for parsing, translation and other NLP tasks. Only a few animacy classifiers and animacy-annotated corpora exist internationally. For Dutch, animacy information is only available in the Cornetto lexical-semantic database. We augment this lexical information with context information from the Dutch Lassy Large treebank, to create training data for an animacy classifier that uses a novel kind of context features.

We use the k-nearest neighbour algorithm with distributional lexical features, e.g. how frequently the noun occurs as a subject of the verb 'to think' in a corpus, to decide on the (predominant) animacy class. The size of the Lassy Large corpus makes this possible, and the high level of detail these word association features provide, results in accurate Dutch-language animacy classification.

1. Introduction

Animacy is a semantic property of nouns that describes whether the referent of the noun is alive or sentient, and to what degree. In recent years, this property has been shown to be a relevant one for natural language processing. It plays a role in various linguistic phenomena across languages, and can be used in determining sentence acceptability and grammaticality. For example, the sentence in (1) would be grammatical if some human being was the noticed thing. *Who* is used to refer to human entities, high on the animacy scale.

(1) ∗ He noticed the table, who was just standing there.

(2) ? The table noticed the woman, who was just standing there.

And while (2) is grammatical, it may not be acceptable. Clearly it has a problem, and it is a problem of felicity. Infelicitous sentences do not violate grammatical rules, but are still unnatural somehow — in this case, because a semantic selection restriction is violated. It does not make sense for tables (or other items lacking sentience) to be the subject of noticing.

Despite the existence of such animacy effects, this property is rarely included in annotation efforts of text corpora and, perhaps for that reason, Natural Language Processing (NLP) tools rarely incorporate animacy in their algorithms. Zaenen et al. (2004) theorize that this is because animacy, in English, often does not influence grammaticality, although it is important for felicity. Because English is the language with the most language resources and corpora, and the language for which the majority of NLP tools is developed, its properties have a strong influence on the design of corpora, annotation schemes or NLP algorithms. Many NLP applications are mainly interested in grammaticality, for which animacy is not an important distinction in English. However, felicity aspects can be important as well, particularly in natural language generation tasks.

Automatically determining the animacy of nouns would allow NLP tools such as parsers to use this property, and allow animacy effects to be studied computationally with large amounts of data. Øvrelid (2009) created an animacy classifier for Swedish, and showed that it can be used to improve parsing accuracy. We are not aware of the existence of any such classifier for Dutch, so we will discuss our solution to Dutch animacy classification.

Automatic animacy classification is the task of deciding which of several animacy-related categories a noun belongs to. The phenomenon of animacy in language is more complicated than what can be captured by the mere distinction between animate or inanimate. There is a wide variety of possible animacy classes, and their category boundaries may be different for different grammatical phenomena. One can also debate whether animacy is a set of classes, a hierarchy, or even a gradient scale. Practical matters also play a large role in this problem, such as the availability of classifier training resources. For supervised learning, some sort of 'gold standard' animacy information is required. Several animacy classifiers have been developed for other languages than Dutch, though they all differ in their method, largely for practical reasons as well. We will examine them and discuss whether aspects of their methods are applicable to Dutch.

In this article, we describe an animacy classifier for Dutch, where the goal is to have a system that can use the linguistic resources that are available for Dutch to provide the best possible classification, given the limitations of the resources. In the future, this classifier can aid in the annotation of Dutch corpora with animacy information, and help NLP tools for the Dutch language to use the animacy property of nouns.

In Section 2 we start with a discussion of animacy and its possible categorizations. Then, we will discuss existing approaches to animacy classification in Section 3. We proceed to describe our methodology and the linguistic resources that we use in Section 4. The evaluation of our system is discussed in Section 5. We discuss applications of our method (Section 6), as well as possible future work in Section 7 and conclude in Section 8.

2. Animacy

The most basic animacy distinction is between animate and inanimate nouns. The category of animate nouns can include personal pronouns, person names, nouns such as *woman*, *participant*, *carpenter*, *dude*, *northerner* and possibly *squirrel* and *angel*. Inanimate nouns can include *fountain*, *second*, *observation*, and possibly *community*, *oak*, and *robot*. Various animacy categorizations and category boundaries are used in linguistic theory and found in languages. In this section, we discuss animacy, its linguistic effects and the ways in which it can be described.

2.1 Animacy hierarchy

To perform a classification task, class labels should be defined. In the case of animacy, this is fairly complicated. Semantically, animacy can be seen as a hierarchy, ranging from a reference to a human (most animate) to a noun that refers to something inanimate. Various categories and subcategories can be defined in between. The choice of categorizations often differs by language, and may change over time. A basic example of such a hierarchy, which first appeared in Silverstein (1976), is HUMAN > ANIMAL > INANIMATE. Some sources also include personal pronouns or personal names in the hierarchy, including them as strongly animate nouns. One such hierarchy is discussed in DeLancey (1981) to explain the system of case marking in some languages. The categories are: 1st & 2nd person > 3rd person > human > animate > natural forces > inanimate.

In cases where animacy plays a role in a linguistic phenomenon, the phenomenon generally applies only to elements above a certain cut-off point in this hierarchy, for example, only to nouns referencing animals or higher animate beings (de Swart et al. 2008). Effects that cannot be explained by a two-way distinction are generally probabilistic or processing effects. In the case marking model

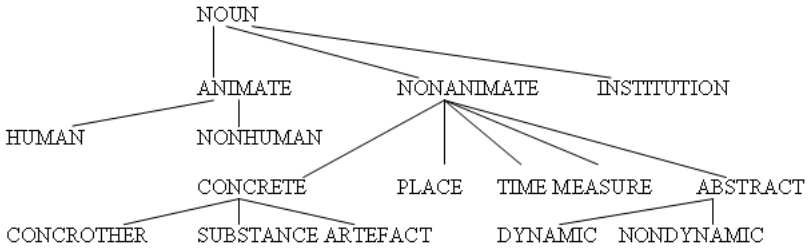


Figure 1: The animacy hierarchy used by the Cornetto lexical-semantic database, from Martin et al. (2005).

discussed by DeLancey (1981), the cut-off point between the use of two kinds of marking may lie between 1st & 2nd person and 3rd person pronouns. However, it has also been argued that the 'person' property should not be conflated with animacy (Comrie 1989).

De Swart et al. (2008) state that the animacy hierarchy should be seen as a gradient, based on the unclear boundaries of categories. In some cases, there are 'grey areas' when animacy categories are observed in language, where nouns that are borderline animate or inanimate may behave in both ways. Such a view lends itself well to probabilistic accounts of animacy, but is not generally adopted in the study of animacy effects on grammar, which are defined in terms of rules that operate on categories. We follow this convention in our classification model.

For some languages the animacy categorization is considered to be both grammatical as well as semantic in nature. For these languages the category of animacy expressing nouns also may include some nouns denoting objects and abstractions (Aissen 1997). Algonquian is an example of a family of languages in which the animacy categorization clearly does not match what an ontology would consider to be animate or inanimate. A survey by Quinn (2001) discusses some examples of nouns that are unexpectedly considered grammatically animate, such as nouns for fluid containers — the Penobscot nouns corresponding to kettle, cup and spoon. This seems to have originated through analogy and convention, and can be compared to grammatical gender which often does not correspond to biological genders of noun referents. The Dutch language, which rarely makes animacy explicit, does not seem to have such differences, but we should keep a conceptual distinction between grammatical animacy and semantic animacy in mind. Even though grammatical animacy may have more interesting applications, our work focuses on classifying semantic animacy — the lack of explicit animacy in Dutch makes it difficult to capture information on grammatical animacy.

While these hierarchies are used in linguistic literature to explain mainly grammatical phenomena, more elaborate animacy hierarchies to describe animacy are also used, based on ontological distinctions. They can be found in language documentation efforts rather than linguistic theories or grammars. The grammatical-semantic distinction is visible here too. An example of such an elaborate hierarchy is the one used for the Dutch Cornetto lexical-semantic database (Martin et al. 2005), which subdivides the inanimate category into various subcategories, as shown in Figure 1.

For animacy, this annotation scheme uses a hierarchy with animate and inanimate categories at the top level, as well as the institution category. The animate and inanimate categories are subdivided further. The scheme is limited in scope but designed with annotation in mind - for example, there are categories like concrother to handle borderline or exceptional cases. Since this hierarchy was designed for a lexical-semantic database, it was likely designed with a semantic notion of animacy in mind, rather than a set of grammatical classes.

In this work, we have adopted a simplified version of this scheme as our classification scheme of choice. Since the Cornetto database was the only large source of animacy information available for the Dutch language, we are limited to using its animacy categories or supersets thereof. Our method

requires such a resource for evaluation and supervised learning. Since distinctions between various types of inanimacy are unlikely to be relevant for grammatical effects or NLP and more difficult to classify from language data, we have generalized the scheme to a three-way classification of HUMAN, NONHUMAN, and INANIMATE, adding INSTITUTION to NONHUMAN since it seemed to contain other institutions. The nonhuman animate class is quite broadly defined, including groups of humans, organizations, foods, plants and vehicles, and unfortunately, there is no further subdivision of this class available. This means we have no alternatives but to use this broad definition of Martin et al. (2005). Our scheme therefore consists of the following three classes:

- HUMAN: Nouns with a human referent.
- NONHUMAN: Nouns with some degree of (biological) animacy, excluding humans.
- INANIMATE: Nouns with an inanimate referent.

Animacy is about the referent of a word, but in some contexts it can be unclear what the referent is (Zaenen et al. 2004). A word may be ambiguous in animacy — the word 'monster' can be used to refer to a creature, but a human can also be called a monster. In general, in figurative language or expressions nouns may have unusual referents. In fictional narratives, a normally inanimate entity may be sentient, behaving more like an animate actor. De Hoop (2012) has studied this, using a Dutch book where the first person narrator is a painting, and comparing it to a book by the same author with an animate narrator. Preliminary results indicate that these sentient inanimate objects behave the way animate entities would behave in language. Local context affects the animacy of the referent — sentience of entities can deviate from reality and this seems to affect the way they are processed as well, showing that animacy is based on semantics even though it can affect a language's grammar.

2.2 Grammatical effects of animacy

There are many ways in which the animacy property can affect grammar in a language. Dahl and Fraurud (1996) provide a non-exhaustive overview:

- Subject and object marking (such as accusative case marking)
- NP-internal case markings (such as the possessive)
- Restriction of subjects of transitive verbs (requiring them to be more animate)
- Hierarchical restrictions (the subject needs to be more animate than the object)

An example of subject and object marking affected by animacy is found in Russian. There, the animacy class of nouns is reflected in their accusative case marking. This marking distinguishes two animacy classes, animate and inanimate. Animate accusative nouns in plural are marked in the same way as the genitive, and inanimate accusative nouns in plural are marked in the same way as the nominative. These examples from Fraser and Corbett (1995) demonstrate the difference:

(3) pervyh (acc=gen) studentov (acc=gen)
first student
'the first students'

(4) pervye (acc=nom) zakony (acc=nom)
first law
'the first laws'

The agreement of the adjective shows the same animacy pattern. Example (3) shows the marking on an animate noun and its adjective; (4) shows an inanimate noun with the same adjective.

In languages with such animacy-based marking, the task of animacy classification can be tackled by simply checking how the nouns are marked in the accusative, if an annotated corpus is available. Unfortunately, Dutch has no such marking, and in fact has very few constructions where animacy is made explicit by the grammar.

One of the few cases where animacy surfaces is the selection of relative pronouns in *wh-cleft* constructions, where a constituent is put into focus by putting it in a dependent clause at the start of the sentence. In other contexts, just noun gender is sufficient to explain the selection of Dutch relative pronouns, but in the case of wh-clefts, animacy also plays a role:

(5) a. **Wat** ik leuk vind, is die tafel(GEN=COMM,-HUMAN)
what i like, is that table

b. **Wat** ik leuk vind, is dat huis(GEN=NEUT,-HUMAN)
what i like, is that house

c. **Wie** ik leuk vind, is dat kind(GEN=NEUT,+HUMAN)
who i like, is that child

d. **Wie** ik leuk vind, is die vrouw(GEN=COMM,+HUMAN)
who i like, is that woman

These constructions occur in English as well, and can be phrased in a similar way. Example (5) shows that in this construction, the relative pronoun does not vary only with gender, as it would if we were to use *d*-pronouns (*die*, *dat*). The animacy property is required to explain the variation in this example, just as in the English equivalents. For a more extensive analysis of this phenomenon that also includes gender effects, see van Kampen (2007); (5) is a constructed example based on her work. The third sentence might need a question mark, and *wat* could also be used there. However, a corpus search[1] shows almost no cases of *wat* being used to refer to animate entities.

Another example, described by de Swart et al. (2008), comes from written Dutch. Some quantifiers such as *meeste* 'most' and *beide* 'both' are marked with a suffix *-n* when they have a human referent (6) but are unmarked in reference to other entities (7):

(6) De studenten hebben beide*(-n) het boek gelezen.
the students have both the book read
'The students have both read the book.'

(7) De boeken werden beide(*-n) door de studenten gelezen.
the books were both by the students read
'Both books were read by the students.'

This effect can also be observed in nominalized adjectives and participles referring to humans.

Both of these Dutch animacy effects divide the animacy scale into a HUMAN and a NONHUMAN category. The Dutch language otherwise lacks clear cases of animacy marking, such as the Russian examples (3, 4). However, not all animacy effects are explicit in the grammar. Sometimes they are merely preferences, or processing effects, which have mostly been discussed in psycholinguistics literature and can be expressed in terms of probabilities.

2.3 Processing effects

A well-known example of a probabilistic effect is called the dative alternation. It has been extensively studied in psycholinguistics, and it is often cited as an example of a syntactic difference without a meaning difference. A transitive verb such as *give* can be phrased as a double object construction (8) or as a prepositional dative (9):

1. Dutch Wikipedia dump from 04-08-2011, automatically parsed with the Alpino parser

(8) He gave his friend the ticket.

(9) He gave the ticket to his friend.

The choice between these syntactic structures is affected by different features such as definiteness and animacy of the referent, though these are merely tendencies, not clear rules. In a study by Bresnan et al. (2007) inanimacy of the recipient in US English has a strong correlation with use of the prepositional dative, and including other features, they are able to predict this syntactic choice in US English correctly with 94% accuracy.

A study by Mak et al. (2002) argued that animacy affects the processing of relative clauses in Dutch. A common finding in psycholinguistic literature has been that subject relative clauses are easier to process than object relative clauses in various languages. Object relative clauses take longer to read for this reason. In subject relative clauses, the relativized element has the subject function in the relative clause (10), while in object relative clauses, it has the object function in the relative clause (11):

(10) **The cat** that touched the apple fell off the table.

(11) **The apple** that the cat touched fell off the table.

Reading times for subject relatives such as in (10) are usually found to be shorter, indicating less processing difficulty.

Mak et al. (2002) have found that this only holds for object relative clauses when the object is animate. It is theorized that readers interpret the animate noun phrase (NP) as the subject, when the two NPs involved in a relative clause differ in animacy. This assumption is likely made because subjects are more likely to be animate entities. This disambiguates them at an earlier stage than relative clauses involving two inanimate NPs, somehow preventing the processing difficulties for object relative clauses from occurring. This finding indicates that animacy plays a role in human sentence processing, guiding the choice of whether a clause should be read as an object or subject relative. This also supports the idea that knowledge of animacy categories may be beneficial for sentence parsing in Dutch.

A related effect can be found in Dutch object fronting. Object fronting is a construction that was found to be more common when the subject is higher on the animacy hierarchy than the object. If object fronting would be used when noun phrases deviate from their 'standard' roles, it would become more difficult to parse the roles correctly, and the message would become less clear. As a result, objects with lesser animacy are more commonly fronted (Bouma 2008).

In the cases discussed here, it would not be ungrammatical to do, for example, animate object fronting, but it would be infelicitous. To produce natural sentences with certain constructions, knowledge of animacy is important. Animacy also plays a role in verb selection restrictions, a sentence may be infelicitous if a senseless argument is used with a verb (*the banana thinks*). This is an animacy effect that our system exploits, and we will address this in Section 4.2.

3. Automatic animacy classification

There has been some previous work on animacy classification for various languages other than Dutch. In this section, we will discuss these classifiers.

3.1 Animacy classification based on morphosyntactic corpus frequencies

For Norwegian and Swedish, an animacy classifier based on frequency counts from a dependency-parsed corpus has been developed, using a basic two-way classification scheme, animate/inanimate (Øvrelid (2004, 2005, 2008, 2009)). The features it uses are syntactic and morphological ones counted over an entire corpus, for example, how often a specific noun occurs as an object. Features of every

instance (token) of a noun are counted, and added up for the noun lemma (type). The classifier is therefore not sensitive to the context of an instance, a property shared by all animacy classifiers developed so far. By taking all contexts into account, much more information is available to base the classification decision on. The classifier originally used decision trees, and in later work used the k-nearest neighbour algorithm as implemented in TiMBL (Daelemans et al. 2007).

Øvrelid's classification features are linguistically motivated and include subject and object roles (subjects are more commonly animate), occurrence in the passive voice (occurs more often with agents as demoted subjects which are more commonly animate), anaphoric reference, reflexivity, possession (genitive case), and in later work, proper nouns, noun gender, number, definiteness, and all syntactic dependency relations a noun may occur in. Of these, subject and object roles are shown to be the most helpful, and are also the most commonly occurring features.

In the latest evaluation (2009), on Swedish, the system reaches 96.8% accuracy for nouns with a frequency of more than 100 in the evaluation corpus (1668 instances). The baseline score for this task, a score that would be obtained by classifying every noun as inanimate, is 90.5%. Øvrelid also shows that the (less common) animate class is more difficult to classify, particularly for lower-frequency nouns. Furthermore, she also demonstrates that the task of dependency parsing may benefit from animacy information by taking a standard language-independent dependency parser and training it on a treebank with and without automatic animacy annotation. The parser that was trained on the animacy-annotated achieved a significantly higher labeled attachment score. This shows that animacy information, even when automatically obtained, may indeed help other NLP tasks.

Most of the linguistic features of this method are also applicable to Dutch (except for a genitive case indicating possession), though since they did not perform well anyway, this is not very interesting. Since there is a larger corpus available for Dutch, we can use more specific forms of the good features, i.e. subject dependencies with specific verbs.

3.2 Animacy classification based on distributional lexical features

Baker and Brew (2010) describe a complex approach to animacy classification in Japanese, that uses various language-specific techniques and heuristics for better coverage of loanwords, one of which is to use the 'person' suffix to group words that refer to people. The core of their method is a Bayesian classifier with an interesting feature-type based on a large data set. They use the frequency with which a noun occurs as a subject (or object) of specific verbs. This means that each verb is a feature, with the values representing the occurrence of the noun and the verb in a subject relation. This can be contrasted with the subjects and objects feature used by Øvrelid. She only counted the number of subject relations in general, while this work counts the number of subject relations for each verb. This feature type is motivated by the fact that verbs often have semantic selection restrictions that can involve animacy — for example, a subject of the verb *to think* is normally sentient, and therefore animate.

For feature values, instead of raw frequency, Baker and Brew use the number of animate subjects for the verb feature, divided by the total number of subjects, as found in the training data. In their evaluation, they obtain 88% accuracy over 36% coverage without additional processing, going up to 95% accuracy and 51% coverage with the suffix grouping, and 88% accuracy with 97% coverage when using an English classifier to cover loanwords. Even though the method is claimed to be multilingual, it performs better on Japanese than on English, due to the language-specific heuristics. Nevertheless, the feature types they used (frequencies of dependencies with specific words) are interesting for any language for which there is enough dependency-parsed data available, such as Dutch.

3.3 Lexical-semantic databases or dictionaries

Orasan and Evans (2007) based their animacy classification on a large lexical-semantic database for English, WordNet. WordNet is organized hierarchically through hypernym and hyponym relations between word senses, also called synsets (synonym sets). At the top of the WordNet hierarchy, there is a small set of generic words known as *unique beginners*. Some of them are related to animacy and large parts of their hyponyms are animate. They use this information to infer the animacy of anything occuring under these unique beginners in the hierarchy. In cases where a noun has multiple senses, some of which are inferred to be animate and others inanimate, the ratio of animate to inanimate senses is computed and a threshold can be set to classify them as either animate or inanimate. They also present an alternative method, in which animacy information is propagated bottom-up; however, this requires an animacy-annotated corpus, which is an unrealistic requirement in most cases.

Orasan and Evans intend to use animacy information for improved anaphora resolution. They apply it by filtering out any candidate referents that do not agree in animacy with the pronoun — for example, *it* cannot be used to refer back to *the man*. Their methodology is based on this, and the anaphora resolution approach shows in their definition of animate NPs, which they consider to be any noun that is referred to using *he*, *she* or related animate pronouns. This contrasts with all of the previous discussion, where animacy categories based on semantics were used. Since English anaphora involve two classes of animacy, an animate/inanimate distinction was used.

De Ilarraza et al. (2002) describe an animacy annotation effort for the Basque language. They needed information about noun animacy to solve some common ambiguities in machine translation to Basque. However, Basque is a fairly under-resourced language, so no lexical-semantic database was available at the time. They instead used the semantic relationships described in an electronic monolingual dictionary to classify a large number of words, starting from a small, manually annotated seed set of 100 nouns. The method is similar to that of Orasan and Evans (2007), and they used synonymy relations in addition to hypernyms and hyponyms. Because of the resource used, they infer hypernymy and hyponymy from dictionary definitions such as:

(12) aeroplane. **vehicle** that can fly

This definition implies, in a structured manner, that an aeroplane is a type of vehicle. Based on the manually annotated set, the method achieves over 99% accuracy with a coverage of 68.2% in the classification of all common nouns in a 1 million word corpus. The method cannot generalize to unknown words.

These lexical-semantics based methods are also applicable to the Dutch language, since such resources are available; however, we believe contextual information is required to have a method that generalizes well, and the use of hypernym and hyponym relations would mean relying on a large amount of manual annotation.

3.4 Lexical pattern on web-scale N-grams

Ji and Lin (2009) built a system for animacy knowledge discovery from Google N-grams Version II, a very large corpus of n-grams gathered from the web, automatically annotated with part-of-speech tags (but not full syntactic trees). They use a simple lexical pattern, based on one of the (few) cases in which animacy affects grammar in English, to obtain animacy information. The pattern exploits the fact that relative pronouns express animacy, and can refer to nouns that occur in the main clause directly before them:

(13) He met the **writer who** wrote the new book.

(14) She saw the **place where** she had dinner yesterday.

The relative pronoun occurs either directly after the noun in the sequence of words, or there is a comma in between, but nothing else. No syntactic knowledge is required to extract this pattern.

Since this only works for one very specific pattern, a lot of data is needed to make this work, but Google N-grams provides this — this pattern occurred 664,673 times. This system was designed to be applied in the task of mention detection, in which animacy also plays a role.

This approach would not work for Dutch. The method uses a language-specific pattern that involves English relative pronouns. This construction does not have an animacy-based distinction in Dutch, and could not be used to detect animacy. There is another possible construction that would work for Dutch, discussed in Section 2.2, but it is far less frequent.

3.5 Semantic animacy classification for English

Bowman and Chopra (2012) discuss an animacy classifier for English that can classify into a more fine-grained set of 10 animacy classes similar to the full hierarchy shown in Figure 1. Unlike our hierarchy, their classes were defined with animacy classification in mind. These classes are particularly specific about what our scheme considers NONHUMAN: they distinguish ORGANIZATION, ANIMAL, MACHINE, and VEHICLE.

Bowman and Chopra (2012) use an animacy-annotated corpus of spoken English as their data. They classify over entire noun phrases, rather than lemma types, making this a form of token-based classification. They use a standard maximum entropy classifier with bag-of-word and POS-tag features (for the noun phrase), as well as subject, object and PP dependencies. In this work the subject and object dependencies are only those of a single NP though, not counted over the entire corpus. These features are their worst-performing, probably due tue to the lack of aggregation, while simple bag-of-words features perform best. This might point to a lack of generalization; no theoretical reason is given why a NP's POS tags or words might influence animacy beyond the fact that the word was labeled that way in the training data. Nevertheless, the system performs well, with an accuracy of 84.90% overall (83.04% with bag-of-words only) and 93.50% over a 53.79% baseline for the simplified task of binary (ANIMATE/INANIMATE) classification.

This approach requires a type of resource that is not available for Dutch, an animacy-annotated corpus, but it does show that a classifier can learn to classify a fine-grained set of classes, if the right resources are available. This work also differs in that spoken language data is used, a part of the Switchboard corpus specifically annotated for animacy, and only NPs with full annotator agreement were considered. They used many inanimate classes which could be confused, but only HUMAN and ANIMAL for animates. which likely biased the data, leading to a larger proportion of human NPs and a lower baseline.

4. Method

We will perform animacy classification with the three classes defined in Section 2.1 — HUMAN, NONHUMAN animate, and INANIMATE. Following most earlier work, this will be a type-based classification. This means we will be working with lemmas from a word list, as opposed to tokens from a corpus. This may introduce ambiguity if a word has two senses with opposite animacy, for example, the Dutch word *monster*, which means 'sample' (inanimate) as well as 'monster' (nonhuman or human animate). An examination of a preliminary version of DutchSemCor, a corpus with word-sense level annotation (Vossen et al. 2012), showed that, of the 2072 nouns annotated with a semantic type, only 34 types (1.5%) are ambiguous in terms of animacy. This indicates that Dutch texts may also have low animacy ambiguity. A similar finding motivated this decision for Swedish in Øvrelid (2009). The type-based approach also allows more information can be extracted for each word.

As in the Swedish animacy classification project of Øvrelid (2009), we make use of the k-nearest neighbour (KNN) algorithm as implemented in TiMBL (Daelemans et al. 2007). This algorithm, also known as memory-based learning, is a supervised machine learning method that compares feature vectors of novel items to those of items for which the class is already known. It then bases its classification decision on the class of the k nearest items (in terms of feature similarity), as well

Frequency	Verb	Construction	Role	Noun
12	schrijf	intransitive	su	Amerikaan (American)
17	schrijf	intransitive	su	artikel (article)
15	schrijf	transitive	su	econoom (economist)
8	schrijf	np_np	su	fan (fan(person))
6	schrijf	transitive	su	hoofdpersoon (main character)
48	schrijf	sbar	su	Le Monde
40	schrijf	intransitive	su	mens (human)
11	schrijf	np_ld_pp	su	Mozart
5	schrijf	sbar	su	Oscar Wilde
1	schrijf	transitive	su	zon (sun)

Table 1: Subject dependency relations of the verb *schrijf* (to write), extracted from the Lassy Large corpus.

as any additional items within the same distance, where k is any number of neighbouring items. KNN is a lazy learning algorithm, preserving all training instances in its 'model', without losing information. This is an important property for handling sparse data issues, which are likely to occur considering the large number of features we use.

4.1 Data

Since no Dutch corpus annotated for animacy was available at the time, the Dutch animacy annotation data from the Cornetto lexical-semantic database was used as gold standard noun data. Recently, DutchSemCor became available for this purpose, but as mentioned in the previous section, animacy ambiguity is uncommon and a word list or dictionary is sufficient information for classifying lemmas that are not ambiguous in animacy. Furthermore, the manually tagged part of this corpus is 'sense-balanced' (all senses of a word occur equally often), making it an unrealistic evaluation gold standard.

A dictionary of nouns and their animacy status was extracted from the Cornetto database. The database consists of 40.392 word senses, but for the dictionary, all senses of the same word (lemma) were merged and duplicates removed, since we will be doing type-based classification. All types where the different senses had different animacy categories, or no animacy value at all, were filtered out. Out of 31.959 types in total, 1008 had multiple different animacy categories (they were ambiguous in terms of animacy). After simplifying the annotation scheme to our three classes, the dictionary consisted of 5.311 nouns labeled as HUMAN, 1.908 NONHUMAN, and 23.732 INANIMATE. The category definitions used in this database mostly seem to have a biological basis, which may not be ideal for many linguistic tasks. As defined earlier, the nonhuman animate class is also very broad, containing things ranging from groups of humans to organizations to foods, plants and vehicles. However, it is the only large-scale animacy-annotated word list available for Dutch, so this is what we have used.

We follow the approach of Øvrelid (2009) and Baker and Brew (2010), and use context features from an annotated corpus as input for our classifier. In order to obtain such features for our classifier, the Lassy Large corpus was used (van Noord et al. 2009). Full syntactic dependency trees, including dependency roles such as 'subject', are present in its annotation. This corpus consists of about 1.5 billion words, and the sentences have been parsed automatically by the Alpino parser for Dutch (van Noord 2006). No human has checked the correctness of these sentence parses, and they may contain some errors. However, this parser is the state of the art for Dutch (Plank and van Noord 2010). The animacy-annotated nouns from the Cornetto data were looked up in the Lassy Large corpus, and specific types of dependency relations in which they appear were extracted, i.e. subject-verb collocations. These were used as context features for our classifier.

As an example, Table 1 shows some subject relation information of the verb *schrijf* (to write), extracted from this corpus. It contains the verb and noun, their role (always *su* - subject, in this case), the construction in which the relation occurs, and the frequency of this relation. In this case, the frequencies are counted separately for each construction (i.e. transitive, instransitive), but we sum them together as we are only interested in roles. The list does not include instances of particle verbs, such as *afschrijven* (to write off), which would be listed under *schrijf_af* in this corpus.

The full list of subject nouns for this verb *schrijf* is dominated by names of people and organizations who have written something, like writers and politicians. It is a word that mostly takes human nouns in the subject role. However, we also find some exceptions, such as 'article' or 'newspaper', which can be explained by constructions like "The article says/writes that ...". In Dutch the verb for writing is used in this sense. There are also several seemingly senseless entries, such as 'sun', but these often have a frequency of 1 which indicates that they might be the result of some sort of parsing error or even a spelling error in the original text — maybe *zoon* (son) was intended for *zon* (sun), or *zo'n* (such a) could have been misspelled and then misparsed as a noun. Since we are using a statistical machine learning method, erroneous low-frequency outliers should not affect the final result much.

4.2 Features

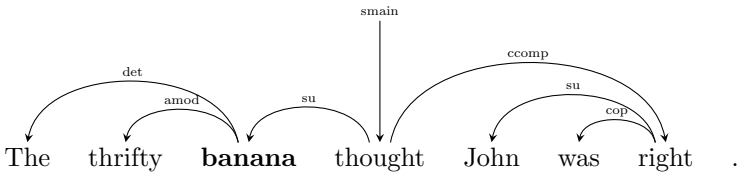
Øvrelid (2009) made use of grammatical features, for example, the ratio of subject vs object roles of a noun. We have also tested this feature, but it did not work (81% accuracy to an 80.92% baseline), possibly because we use three animacy classes instead of two. A more detailed approach to context information was taken by Baker and Brew (2010). Rather than counting noun subject/object roles in general, they counted noun subject/object roles with specific verbs, i.e. how often *driver* is the subject of *to drive*. Each common verb is a feature, and values derived from those frequency counts are the feature values. They used a Verb Animacy Ratio, the overall ratio of animate subjects (or objects) used with a verb in the corpus. This ratio was used as their feature value, rather than just the total frequency of occurrence of the subject-verb relation. However, we foresee some difficulties with generalization to novel data with this approach. The animacy ratio will always be based on the training data, even if the animacy ratio of verbs in a different corpus or document is completely different, and the animacy ratio of verbs cannot be known without actually having animacy annotation. We also question the usefulness of relying on knowledge of 'verb animacy' for the task of determining a noun's animacy. Their idea of verb animacy is basically knowledge of selection restrictions, and selection restrictions must be derived from noun animacy. We would prefer to avoid such additional data requirements. Therefore, we only use dependency frequency counts and statistics derived from them. Since counts are needed for each verb this approach requires a large amount of data, which can be found in the Lassy Large corpus for Dutch.

The effectiveness of this type of feature can be explained by theories of distributional semantics. According to this view, words that are similarly distributed (i.e. occurring in similar contexts), also have similar meanings. This idea was explored by Hindle (1990), who used a similar methodology of examining subject and object relations with verbs for semantic noun classification (in terms of semantic similarity). He defines noun similarity in terms of the mutual information (a measure of association) of verbs and their (noun) arguments. Our approach follows this basic idea, and we also use association measures as feature values to determine which verbs (or adjectives) are typical for a noun, rather than a Verb Animacy Ratio.

The following example will illustrate this:

Noun	*kom_binnen*	*luid*	Animacy
bestuurder	0.0000027447	0.9999999999	human
parool	0.7301989638	0.0002008943	nonanimate
VVV	0.4287437089	0.4504258805	nonhuman

Table 2: Example feature vector for KNN animacy classification. Contains three training nouns: *bestuurder* (driver, director), *parool* (parole (word of honour), a newspaper), *VVV* (tourist information offices).



This sentence contains the inanimate noun *banana*; however, it's not a felicitous sentence. If this sentence said *trader* instead of *banana* it would be fine, but as it is, it violates the semantic selection restrictions of the adjective *thrifty* and the verb *to think*. These selection restrictions are generally not hard rules, but they are certainly tendencies that can be identified from frequencies of such relations in a corpus. The verb *to think* is much more likely to take highly animate and sentient subjects (i.e. humans). Thus, it is possible to make inferences about a word's animacy based on the dependency relations it occurs in, in a large corpus. We have used subject, object and adjective dependency relations in this way.

After feature extraction, we may end up with a feature vector with values that express some association, such as the highly simplified example of Table 2. Here, nouns would be classified based on their subject relations with two specific verbs, *binnenkomen* (to come in) and *luiden* (to sound/say/read). The values represent the statistical association strength between each noun and the feature verbs, i.e. whether they occur together more often than would be expected by chance, in this case computed with Fisher's Exact Test.

The classifier then stores this data in memory, and this is the model. No generalization takes place. New data items, for which the animacy class is not known, can then be compared to this model to classify them. Figure 2 illustrates the concept. It shows a two-dimensional feature space of the two verb-subject relations. Normally there would be more verb-subject relation features and therefore more dimensions. Training nouns are positioned in this space depending on their association strength with the verbs. For these training nouns, the class is known — the black dots represent inanimate nouns, the light grey ones are human animates, and the dark grey ones are nonhuman animates. A new item (white) can be compared to one or more (1-k) *neighbours* in the feature space, hence the name of the algorithm. In the example, for the novel word *dracht*, the nearest five neighbours, and any other neighbours on the circle, in this simplified two-dimensional feature space are taken into account. We see that the single nearest neighbour is a human animate word, but this must be an outlier or unusual word, since the remaining 4 of the 5 nearest neighbours are known to be inanimate, and thus our novel word is classified as inanimate.

This process involves several measures, for which TiMBL provides a few options. A similarity measure is needed to measure the distance between a noun and its neighbours in the feature space, i.e. how similar they are. Furthermore, features that are information-theoretically more informative are weighted more heavily, which requires an information metric. If $k > 1$ neighbours are used, class voting also comes into play — how much each neighbour contributes to the classification decision. These options may also take parameter settings, leading to over 900 possible parameter combinations. Rather than manually picking metrics and parameters based on our data, we use

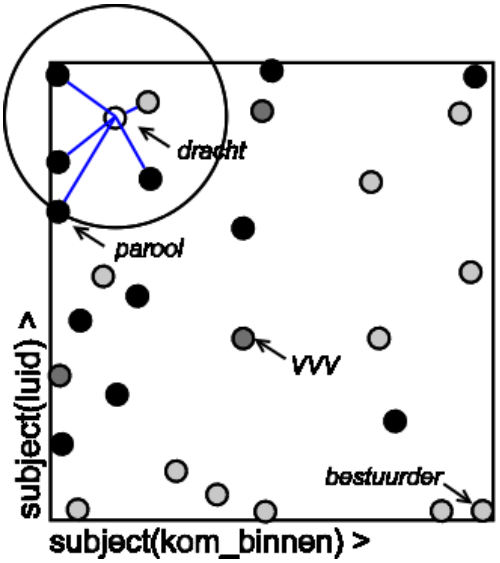


Figure 2: Visualization of a two-dimensional feature space for KNN animacy classification with k=5

Wrapped Progressive Sampling, a method developed along with TiMBL for automatically picking optimal settings. Van den Bosch (2004) has shown that an average of 31.2% error reduction could be achieved on various classification problems by automating this process, compared to using the default settings. However, the process is not guaranteed to choose the best settings.

5. Evaluation

For evaluation, we applied ten-fold cross validation to the entire gold standard set (after frequency cutoffs), in order to obtain an accuracy score that is an average of ten tests, which is more reliable than using just a single test.

5.1 Baseline

Since animacy classification for Dutch has not been done before to our knowledge, and it is difficult to compare the scores to other animacy classifiers that work on different data sets or use a different evaluation method, we start from a very simple baseline. Our baseline is the accuracy score that can be obtained by classifying every noun as the majority class (normally INANIMATE). In the full data set, 76.68% of the nouns are inanimate, but this proportion changes when different frequency cutoffs are used. Unless otherwise noted, we use a frequency cutoff of 10 to exclude nouns that occur rarely or never in the Lassy Large corpus. So to be included in the dataset, a noun must occur in a relevant dependency relation (that we use as a feature) 10 times or more. Low-frequency nouns are very difficult to classify, since there is not much feature information available for them. The baseline accuracy for that set is 80.92%.

5.2 Features

To determine whether we chose the right feature types, in particular *Adjective* relations which have not been used in any previous work, we evaluated each of them individually, i.e. we built a classifier based only on adjective relations. The results are shown in Table 3. The cutoffs were empirically

Feature type	Frequency cutoff	Number of verbs	Accuracy
Baseline			80.92%
Subject rels	25000	86	91.14%
Object rels	100000	18	**92.93**%
Adjective rels	50000	69	88.99%

Table 3: The best classifier results of each dependency feature type

Cutoff-subj	Nr-subj	Cutoff-obj	Nr-obj	Cutoff-adj	Nr-adj	Accuracy
25000	86	100000	18	50000	69	90.13%
25000	86	50000	39	50000	69	**92.68**%
50000	39	100000	18	50000	69	**92.76**%
50000	39	50000	39	50000	69	**92.27**%

Table 4: Classifier results for different object/subject/adjective feature proportions

selected. It shows that each of the feature types performs well above the baseline, even the adjective features, though with slightly lower accuracy than the others.

Next, we trained a classifier with the three feature types combined, which initially performed worse than a classifier based on only object relations. We experimented by varying the cutoffs for each feature type, and found that balancing the feature types indeed improved the combined classifier performance (Table 4).

5.3 Feature values

As noted earlier, our approach is related to distributional semantics, which shows that mutual information can be used to quantify the association between nouns and their dependencies. However, other association measures are available. We evaluated the classifier with various measures for the feature values, the result of which is shown in Table 5.

Binary simply measures occurrence of a relation. **Normalized frequency** normalizes for general noun frequency. **Pointwise Mutual Information** is the measure from information theory used by Hindle (1990), and **Fisher's Exact Test** is an alternative that has some properties beneficial for working with natural language data. It has the advantage of being exact and does not make assumptions about the distribution of the data, which tends to be skewed in NLP tasks. For these reasons it has been used in similar tasks involving association of lexical items. However, this method does not stand out in the results. A possible explanation could be that this test does not measure effect size - it can say that the lexical items are dependent or independent, but is not really designed

Feature metric	Accuracy
Baseline	80.92%
Binary	92.19%
Frequency	**93.09**%
Normalized Frequency	89.23%
Pointwise Mutual Information	**92.27**%
Fisher's Exact Test	91.37%

Table 5: Comparison of different association measures for features of the classifier

Frequency cutoff	Number of nouns	Baseline	Accuracy
0	30.950	76.68%	84.14%
1	16.454	78.16%	90.82%
10	12.168	80.92%	**92.27**%
100	6.276	84.00%	91.06%
1000	1.671	88.99%	88.62%

Table 6: Classifier performance on datasets with different frequency cutoffs

to specify by how much. But in fact, the PMI, raw frequency, and the binary measure surprisingly all result in good accuracies. Nevertheless, we have used PMI for our experiments, since there are good theoretical reasons for using an association measure, and such measures are widely used in linguistics, i.e. studying collocations. PMI can express whether the verb and argument co-occur more often than would be expected by chance, and by how much.

5.4 Frequency effect

In our experiments so far, we have simply chosen to use a dataset of all nouns with a frequency of 10 or more, since nouns with fewer instances than that will be very difficult to classify due to sparse data (such nouns only occur with up to 10 features at most). However, for some applications it may be interesting to examine other data sets as well. In the evaluation of her animacy classifier, Øvrelid (2009) included results for different *accumulated frequency bins*, i.e. all instances above a certain threshold of frequency. She indeed observes better accuracy on data that is less sparse. Our version of this experiment is shown in Table 6. We apply the frequency cutoff to both the training data and the testing data, i.e. we are training a classifier for the task of classifying high-frequency words. In this, we follow the methodology of Øvrelid (2009).

Because we are using one resource to get our animacy dictionary and a different resource to get the feature values, it may be the case that words from the dictionary are unknown in the corpus (or at least do not occur in any relevant dependency relations), and thus no training or testing data can be obtained. These words thus cannot be defined in terms of dependency features, and therefore cannot be tested with at all. This is different from the problem of classifying an unknown word, which does not occur in the training data but shows up in the test data. This explains the difference between the 0 and 1 frequency bins. The difference is so large because we have not applied any sort of preprocessing or normalization between the two data sets. We also find an unexpected drop-off for the highest frequency cutoff, it may be caused by a smaller data set (i.e. training + testing) in general.

5.5 Class confusion

An overall accuracy score of 92.5% still includes many mistakes, especially when the baseline is 80%. This is also indicated by a low macro-averaged F-score of 0.65, which indicates that one of the component F-scores (i.e. the score on one of the classes) is very poor. A major cause of errors is revealed when we examine the confusion matrix of this classifier, in Table 7, which plots the class predicted by the classifier against their real class as stated in the animacy dictionary.

It reveals that this classifier has not learned the properties of the NONHUMAN class, and rarely even attempts to classify any noun as nonhuman animate. It only does so 17 times, 14 of which are wrong, an accuracy of 18%. The HUMAN class is predicted with 87% accuracy, and the large INANIMATE class is predicted correctly 98% of the time. In Section 4.1 we discussed the definition of the classes, and the NONHUMAN class was indeed somewhat vaguely defined, including words that

Confusion	Human	Nonhuman	Inanimate
Human	**153**	1	21
Nonhuman	1	**3**	51
Inanimate	7	13	**966**

Table 7: Confusion matrix of the classifier with optimal settings. Columns are classes predicted by the classifier, and rows are actual noun classes. **Bold** values are correct predictions.

Classifier	Accuracy
Baseline	80.92%
All	**94.00**%

(a) Inanimate – ¬Inanimate

Classifier	Accuracy
Baseline	85.57%
All	**97.45**%

(b) Human – ¬Human

Table 8: Performance of two kinds of two-way animacy classifiers.

are likely to occur in animate contexts such as animals and groups of humans, as well as words that refer to entities that are only vaguely biologically animate, but are unlikely to be very agentive linguistically. This particular class may be problematic or even impossible to infer from the features that we use. However, the Cornetto lexical-semantic database and maybe DutchSemCor are the only large animacy-annotated resource (sharing the same semantic annotation scheme), so we are limited to using the categories that they defined. We performed a few additional experiments to investigate this issue.

5.6 Two-way classification

There is something to be said for an animacy classifier that decides only between ANIMATE and INANIMATE. After all, animacy effects in linguistics generally divide the hierarchy into two classes around some cutoff point, and if the noun is on the animate side, we use one construction, and if it is on the inanimate side, we use the other (i.e. *who* and *which*). However, we are limited by the available animacy data from the Cornetto database. It divides the hierarchy into HUMAN and NONHUMAN ANIMATE, as well as INANIMATE, leaving us with two sensible groupings. We have performed these classification tasks. INANIMATE – ¬INANIMATE simulates an animate-inanimate distinction, though we have seen that the NONHUMAN class is too broadly defined. HUMAN – ¬HUMAN is the distinction that is relevant for the Dutch grammatical animacy effects that we discussed earlier.

Table 8 shows the results. We can see that the HUMAN – ¬HUMAN classifier performs very well, even though the baseline is also high. The performance of the INANIMATE – ¬INANIMATE classifier is comparable to our three-way classifier, even though the task is easier. These results imply that the NONHUMAN class, even though it is described as animate, is closer to the INANIMATE class, at least as far as its verb and adjective dependencies are concerned.

5.7 Data balancing

To gain insight into the cause of the NONHUMAN classification problem, we performed another experiment. While the definition of the NONHUMAN class is vague, another problem is that this class is only a small minority of the data, only around 5% depending on the cutoffs used. That might be too little data proportionally. So, we created another data set, taking all 566 nonhuman nouns, and adding an equal number of random human nouns and random inanimate nouns. In this set, each

Classifier	Accuracy
Baseline	39.91%
Balanced	72.14%

Table 9: Classifier performance on a data set where each noun animacy class is equally frequent (the classes are balanced).

class thus makes up 33% of the data (before frequency cutoffs) and no class is underrepresented. This also lowers the baseline significantly, since the majority class is now proportionally smaller.

The results in Table 9 show that the classifier clearly outperforms the baseline, although the result is not great. The confusion matrix reveals that accuracy on the HUMAN class is 69%, for INANIMATE it is 80%, and for NONHUMAN it is 65%. These accuracies are proportional to the size of each class after frequency cutoffs: 32% for HUMAN, 40% for INANIMATE, and 28% for NONHUMAN. This classifier is now able to distinguish all three classes, although at a reduced accuracy (possibly due to the smaller data set). This result shows that it is possible to identify the NONHUMAN ANIMATE class from these features, the machine learning algorithm just does not handle the lack of instances of this class well.

6. Applications

While animacy annotation has many uses, we have to be aware of the limitations of the annotation that this classifier can provide. For example, there is not much we can change about the definition of the three animacy classes, unless a different animacy resource with a different annotation scheme is created, or a method for reducing the classifier's resource requirements is developed. We also have to keep in mind that automatic classification is never perfect, and use of this classifier's output in another NLP system will introduce errors. That may or may not be a problem depending on the application. This classifier is in turn also based on automatically annotated data with the potential of containing errors (the Lassy corpus) but the fact that the data set is much larger than what could be obtained by manual annotation makes up for that. Nevertheless, there are clearly some interesting applications, as evidenced by the fact that most previous animacy classifiers were developed with some specific application in mind.

Our method could be used to annotate more resources with animacy information, in a less labour-intensive way than manual annotation. If the corpus is to be used for linguistic research, a higher level of accuracy than what our classifier has achieved would probably be desired. This can be done by resorting to semi-automatic annotation. In this procedure, animacy annotation is first generated automatically, and is then checked by an expert. However, our three-class animacy hierarchy would also be a limiting factor for some applications. As we have seen in Section 2.1, semantically annotated corpora that are used for linguistic research use more fine-grained animacy hierarchies which subdivide the three classes further.

If the resource is to be used for statistical natural language processing tasks, i.e. to improve disambiguation accuracy in statistical parsing or fluency in text generation, the error rate may not be such a problem, as was the case for the Lassy corpus when we trained our classifier on it. However, structural errors would be problematic, since they would skew the statistical model too much. An example of such an error is the inability of our classifier to learn about the NONHUMAN class, which is classified with 0% accuracy in some cases. For this kind of application, one would have to use the classifier trained on a balanced data set as we demonstrated in the previous section, or a two-class version of the classifier that avoids this issue. The NONHUMAN class may not be useful for some applications, so this could be an acceptable compromise in those cases.

Øvrelid (2009) proposed to incorporate animacy information into a parser, and evaluated whether the output of her own animacy classifier would contribute to better parsing accuracy for the MALT dependency parser, reporting a small but significant improvement in the Labeled Attachment Score of the parser. One way in which animacy may aid parsing is during disambiguation, i.e. a choice of two alternative parses involving a known animate noun, in one parse it is the object and in the other parse, it is the subject. Animate nouns are far more likely to be the subject in most cases.

De Ilarraza et al. (2002) were motivated to perform animacy classification by ambiguity problems in machine translation to the Basque language, where a common preposition is ambiguous when translated from Spanish, and the animacy property of the head is needed for disambiguation. This is a specific situation, but similar ambiguities exist in Dutch, for example when translating the Dutch word *die* in this sentence:

(15) De man **die** op de tafel stond.
The man **who** on the table stood.
'The man **who** stood on the table.'

(16) De stoel **die** op de tafel stond.
The chair **which** on the table stood.
'The chair **which** stood on the table.'

We need to know that *man* is animate in (15) in order to produce *who*, the correct English translation of *die* in this case, rather than *which*, as required in (16). Relatedly, Orasan and Evans (2007) developed their animacy classifier to aid in anaphora resolution, where *he/she* and *it* have an animacy distinction.

Another aspect of machine translation is felicity, i.e. generating 'natural' sounding sentences, which relates to probabilistic animacy effects. Choosing the more common construction of the dative alternation (prepositional dative or double object construction), which is affected by animacy, improves the felicity of a generated text. Grammar correction is another NLP problem that could benefit from animacy information, to the extent that the target language has explicit grammatical effects of animacy.

It should be noted that potential applications are not necessarily limited to the Dutch language. Like most statistical methods, the method we use is fairly language-independent. It should work for any language in which some sort of dependency relations that place semantic selection restrictions on nouns can be identified. The main issue is to find the right resources that can be used for training. For languages that are rich in resources, such as English, this should not be a problem, though for most languages of the world, there are no readily available lexical-semantic databases. Our method requires two resources. Firstly, some sort of dictionary with animacy information in it, though not necessarily a hierarchy of synsets. This dictionary could be extracted from a lexical-semantic database. Secondly, a source of dependency relations. i.e. a dependency-parsed corpus with labeled roles.

7. Future work

The classifier could be improved by reducing its dependence on linguistic resources, for example, by starting with only a small seed set of nouns with known animacy, and using a self-training process. Somewhat related to this is the possibility of doing preprocessing to make optimal use of the available data. Since such preprocessing generally involves language-specific rules, we have refrained from it, but it could be a good way to improve the classifier for Dutch. Baker and Brew (2010) succesfully used several techniques for Japanese. They made use of the compound morphology of Chinese loanwords in Japanese to make the classification task easier. They grouped them by their suffix: for example, *firemen*, *salesmen* and *weathermen* are all types of men. It may be interesting to try some compound splitting heuristics for preprocessing in Dutch.

The possibility of token-based, rather than type-based, animacy classification could also be explored, by means of the recently developed word-sense annotated corpus of Dutch, DutchSemCor (Vossen et al. 2012). Only one such classifier, described in recent work by Bowman and Chopra (2012), exists at the time of writing. We have not compared different machine learning algorithms for this task, as Øvrelid (2009) found no performance difference comparing Support Vector Machines to the KNN algorithm, but this is another possibility to look into.

8. Conclusion

In this work, we have presented the first animacy classifier for Dutch. We have discussed the complexities of animacy and shown various examples of its effects in different languages. We have discussed animacy hierarchies and annotation schemes, as well as existing work on animacy classification for other languages. We then described our own approach to animacy classification, adapted to the state of the art of Dutch linguistic resources, and evaluated the resulting classifier system.

We used a k-nearest neighbour classifier as implemented in TiMBL to classify nouns into three classes of animacy — HUMAN, NONHUMAN ANIMATE and INANIMATE. Due to the availability of the Lassy Large corpus for Dutch, a large automatically annotated corpus, we were able to use novel kinds of distributional lexical features, i.e. dependency relations with specific words, to classify the nouns into animacy classes. Words can impose semantic selection restrictions on their dependencies, for example, the verb *to think* mostly takes animate subjects. By quantifying such subject, object and adjective dependency relations of words for which the animacy is known, the classifier can gain information about such restrictions or tendencies related to animacy classes, and generalize this to novel words to predict their animacy class. We determine whether dependency relations are significant through measures of statistical association, the same method that is used for finding word collocations in general.

We have demonstrated a classifier accuracy of 92-93% on the task summarized above. However, this classifier has problems learning to distinguish the NONHUMAN category, so we have also explored some methods of avoiding that issue — using a different mix of training data, or simplifying the task to two-class classification. In classifying with a two-class scheme of HUMAN-NONHUMAN (other), we reached an accuracy of 97-98%. Lastly, we discussed some potential improvements and applications of the method, such as corpus annotation or use in other NLP tools.

References

Aissen, J. (1997), On the syntax of obviation, *Language* pp. 705–750, JSTOR.

Baker, K. and C. Brew (2010), Multilingual animacy classification by sparse logistic regression, *Ohio State University Working Papers in Linguistics* p. 52.

Bouma, G.J. (2008), *Starting a Sentence in Dutch: A corpus study of subject-and object-fronting*, PhD thesis, University of Groningen.

Bowman, Samuel R and Harshit Chopra (2012), Automatic animacy classification, *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop*, Association for Computational Linguistics, pp. 7–10.

Bresnan, J., A. Cueni, T. Nikitina, and R.H. Baayen (2007), Predicting the dative alternation, *Cognitive foundations of interpretation* pp. 69–94, Amsterdam: KNAW.

Comrie, B. (1989), *Language Universals and Linguistic Typology: Syntax and Morphology*, University of Chicago Press.

Daelemans, W., J. Zavrel, K. van der Sloot, and A. van den Bosch (2007), TiMBL: Tilburg Memory-Based Learner, *Version* **6**, pp. 07–03.

Dahl, O. and K. Fraurud (1996), Animacy in grammar and discourse, *Pragmatics and Beyond, New Series* pp. 47–64, John Benjamins Publishing Co.

de Hoop, H. (2012), Animacy in narratives: the effect of an inanimate narrator on verbs and thematic roles, Presented at Workshop Early Language, University of Groningen.

de Ilarraza, A.D., A. Mayor, and K. Sarasola (2002), Semiautomatic labelling of semantic features, *Proceedings of the 19th International Conference on Computational Linguistics-Volume 1*, Association for Computational Linguistics, pp. 1–7.

de Swart, P., M. Lamers, and S. Lestrade (2008), Animacy, argument structure, and argument encoding, *Lingua* **118** (2), pp. 131–140, Elsevier.

DeLancey, S. (1981), An interpretation of split ergativity and related patterns, *Language* pp. 626–657, JSTOR.

Fraser, N.M. and G.G. Corbett (1995), Gender, animacy, and declensional class assignment: A unified account for Russian, *Yearbook of Morphology* **1994**, pp. 123–150.

Hindle, D. (1990), Noun classification from predicate-argument structures, *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 268–275.

Ji, Heng and Dekang Lin (2009), Gender and animacy knowledge discovery from web-scale n-grams for unsupervised person mention detection, *Proc. PACLIC2009*.

Mak, W.M., W. Vonk, and H. Schriefers (2002), The influence of animacy on relative clause processing, *Journal of Memory and Language* **47** (1), pp. 50–68, Elsevier.

Martin, W., I. Maks, S. Bopp, and M. Groot (2005), RBN-documentatie, *Report, TST Centrale*.

Orasan, C. and R. Evans (2007), NP animacy identification for anaphora resolution, *Journal of Artificial Intelligence Research* **29** (1), pp. 79–103, American Association for Artificial Intelligence.

Øvrelid, L. (2004), Disambiguation of syntactic functions in Norwegian: Modeling variation in word order interpretations conditioned by animacy and definiteness, *in* Karlsson, Fred, editor, *Proceedings of the 20th Scandinavian Conference of Linguistics*. http://www.ling.helsinki.fi/kielitiede/20scl/proceedings.shtml.

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

Øvrelid, L. (2009), Empirical evaluations of animacy annotation, *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

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

Plank, B. and G. van Noord (2010), Dutch dependency parser performance across domains, *Proceedings of the 20th Meeting of Computational Linguistics in the Netherlands*, pp. 123–138.

Quinn, C. (2001), A preliminary survey of animacy categories in Penobscot, *Papers of the 32nd. Algonquian Conference*, pp. 395–426.

Silverstein, M. (1976), *Hierarchy of Features and Ergativity*, Humanities Press.

van den Bosch, A. (2004), Wrapped progressive sampling search for optimizing learning algorithm parameters, *Proceedings of the 16th Belgian-Dutch Conference on Artificial Intelligence*, pp. 219–226.

van Kampen, J. (2007), Relative agreement in Dutch, *Linguistics in the Netherlands* **24** (1), pp. 112–124, John Benjamins Publishing Company.

van Noord, G., G. Bouma, F. Van Eynde, D. de Kok, J. van der Linde, I. Schuurman, E. Tjong Kim Sang, and V. Vandeghinste (2009), Large scale syntactic annotation of written Dutch: Lassy, *Essential Speech and Language Technology for Dutch*, Springer Berlin Heidelberg, pp. 147–164.

van Noord, Gertjan (2006), At last parsing is now operational, *TALN06. Verbum Ex Machina. Actes de la 13e Conference sur le Traitement Automatique des Langues Naturelles*, pp. 20–42.

Vossen, Piek, Attila Görög, Rubén Izquierdo, and Antal van den Bosch (2012), DutchSemCor: Targeting the ideal sense-tagged corpus, *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC2012), Istanbul, Turkey*, pp. 584–589.

Zaenen, A., J. Carletta, G. Garretson, J. Bresnan, A. Koontz-Garboden, T. Nikitina, M.C. O'Connor, and T. Wasow (2004), Animacy encoding in English: Why and how, *Proceedings of the 2004 ACL Workshop on Discourse Annotation*, Association for Computational Linguistics, pp. 118–125.