Computational Linguistics in the Netherlands Journal 3 (2013) 2-18 Submitted 05/2013; Published 12/2013

Semantic classification of Dutch noun-noun compounds A distributional semantics approach

Ben Verhoeven BEN.VERHOEVEN@UANTWERPEN.BE
Walter Daelemans WALTER.DAELEMANS@UANTWERPEN.BE

CLiPS - Computational Linguistics Group
University of Antwerp
Prinsstraat 13, Antwerp, Belgium

Abstract

This article describes the first attempt to semantically analyse Dutch noun-noun compounds using the distributional hypothesis, which states that the semantics of a word is implicitly represented by the words in its context. The purpose is not only to classify compounds based on their semantics. We also investigate in what circumstances this classification works best. Using Ó Séaghdha (2008) as a source of inspiration, a list of 1,802 noun-noun compounds was collected and annotated. The annotators had an annotation scheme and guidelines available with six specific semantic categories (BE, HAVE, IN, ACTOR, INST, ABOUT) and five categories for less specific categories or incorrect compounds. An inter-annotator agreement of 60.2% was found on a 500 compound subset. The task of automatically analysing compound semantics was framed as a classification task for which we can use supervised machine learning algorithms. The instance vectors were created by concatenating the vectors containing co-occurrence information on the compound constituents. In certain variants of the experiment, principal component analysis (PCA) was used as a means of reducing the dimensionality of the dataset. Support vector machines and instance-based learning were used for the machine learning experiments. A maximum F-score of 49.0% was reached on the normal bag-of-words (BOW) data using the SVM algorithm. The PCA data yielded a maximum F-score of 45.2%. These scores should be compared with a most frequent class baseline of 29.5%. The achieved results in both main variants significantly outperform this baseline.

1. Introduction

A notable obstacle for natural language understanding is the productivity that a language exhibits in creating new words. An important and very productive word formation process, at least in Germanic languages, is compounding (Booij 2002, 141). Compounding is different from word formation by derivation in that derivations can easily be analysed by reducing the word to its stem and derivation morphemes. A derivation is merely a syntactic variation of the word stem, with a transparant meaning. In compounding, however, word stems are combined to new words and we do not know the semantic relationship between the two word stems. Since most of these new words are not available in a machine-readable dictionary and their meanings are hence not explicated, a computational system will have trouble interpreting the meaning of these words. Existing NLP applications, such as question answering, information extraction and machine translation systems, will benefit from better compound understanding. Being able to paraphrase a compound and then translate it, is essential for a machine translation system (Nakov 2008). For example, if the system cannot analyse *Antwerp hostel* to mean 'hostel in Antwerp', it could not easily be translated to the French *auberge à Anvers*.

This paper presents initial results on the development of a semantic analyzer for Dutch noun-noun compounds. The structure of this paper will be as follows. First, a summary of related research on the topic will be presented in Section 2. This summary will focus on the methodology used in our own research. This includes a description of classification schemes for annotation and the kind of features used in our experiments. We then describe our annotation scheme and process for the

Dutch noun-noun compounds in Section 3. In Section 4, the classification experiments are discussed, after which we present our results in Section 5. Finally, we posit our conclusions and propose some directions for further research.

2. Related research

This paper builds on some of our own recent work on Dutch (Verhoeven 2012, Verhoeven et al. 2012). Past research on semantic analysis of noun-noun compounds has focused almost exclusively on English, although there are also recent initiatives for German (Hinrichs et al. 2012), Afrikaans (Verhoeven et al. 2012), and some other languages. The problem of semantically analyzing compounds was mostly considered a supervised machine learning problem. Different approaches were proposed considering two main characteristics of the research: the scheme of categories being used for the semantic classification of the compounds, and the features that the machine learning algorithm uses to classify the compounds.

2.1 Classification schemes

Several attempts have been made in the past to come up with appropriate classification schemes for noun-noun compound semantics. These schemes are mainly inventory-based in that they present a limited list of predefined possible classes of semantic relations a compound can have. Early work in computational research is due to Warren, quoted by Rosario and Hearst (2001). Other early birds are Finin (1980) and Lauer (1995).

In some cases, proposed classes are abstractly represented by a paraphrasing preposition (Lauer 1995, Girju et al. 2005, Lapata and Keller 2004). For example, all compounds that can be paraphrased by putting the preposition 'of' between the constituents belong to the class OF, e.g. a 'car door' is the 'door of a car'. Another possibility is using predicate-based classes where the relations between the constituents are not merely described by a preposition but by definitions or paraphrasing predicates for each class. The class AGENT would contain compounds that could be paraphrased as 'X is performed by Y' (Kim and Baldwin 2005), e.g. 'enemy activity' can be paraphrased as 'activity is performed by the enemy'. Different schemes vary from 9 to 43 classes with kappa scores for inter-annotator agreement between 0.52 to 0.62 (Ó Séaghdha 2008, Rosario and Hearst 2001, Nakov 2008, Moldovan et al. 2004, Tratz and Hovy 2010, Barker and Szpakowicz 1998, Wijaya and Gianfortoni 2011).

So far, classification schemes have focused only on noun-noun compounds. It is only recently that other nominal compounds (with verbs, adjectives, adverbs, quantifiers and prepositions as left-hand constituents) are taken into account (Verhoeven and van Huyssteen 2013).

2.2 Features

With regard to the information used by the classifier to assign the classes to the compounds, two main roads are available, viz. taxonomy-based methods, and corpus-based methods. Taxonomy-based methods (also called semantic network similarity (Ó Séaghdha 2009)) base their features on a word's location in a taxonomy or hierarchy of terms. Most of the taxonomy-based techniques use WordNet (Miller 1995) for these purposes; especially the hyponym information in the hierarchy is used. A bag of words is created of all hyponyms and the instance vector contains binary values for each feature (the feature being whether the considered word from the bag of words is a hyponym of the constituent or not). Kim and Baldwin reached an accuracy of 53.3% using only WordNet (2005). Other research was based on Wikipedia as semantic network (Ó Séaghdha and Copestake 2007) or the MeSH hierarchy of medical terms (Rosario and Hearst 2001).

Corpus-based methods use co-occurrence information of the constituents of the selected compounds in a corpus. The underlying idea - the distributional hypothesis (Harris 1968) - is that the

set of contexts in which a word occurs, is an implicit representation of the semantics of this word (Ó Séaghdha and Copestake 2007). This information can be used in different ways. Ó Séaghdha (2008) describes measures of lexical similarity and relational similarity.

The lexical similarity measure assumes that compounds are semantically similar when their respective constituents are semantically similar. The co-occurrences of both constituents will be combined to calculate a measure of similarity for the entire compound. Accuracies of 54.9% (Ó Séaghdha 2007, Ó Séaghdha and Copestake 2007) and 61.0% have been reached (Ó Séaghdha 2008, Ó Séaghdha and Copestake 2008).

The relational similarity measure assumes two pairs of constituents to be similar if the contexts in which the members of one pair co-occur are similar to the contexts in which the members of the other pair co-occur (Ó Séaghdha 2008, 118). Ó Séaghdha and Copestake (2007) report an initial accuracy of 42.3%. This result was improved to 52.6% by Ó Séaghdha (2008). Lapata and Keller (2004) report an accuracy of 55.7% with web-based relational similarity. Their corpus-based similarity's accuracy was only 27.8%.

Nastase et al. (2006) extract grammatical collocations of the constituents from a corpus and use it as features for the classifier. This collocation includes words that appear with the target word in a grammatical relation, e.g. subject, object, etc. Corpus-based and taxonomy-based methods have also been combined by several researchers. Accuracies of 58.3% (Ó Séaghdha 2007), 79.3% (Tratz and Hovy 2010) and even 82.5% (Nastase et al. 2006) were reported.

3. Annotation

The current section will deal with the process of annotation that enabled us to gather the required data for our automatic compound classification experiment. Since we are performing a supervised learning experiment, we need information on the semantics of the Dutch compounds that our machine learners can use for training. This need for a description of the semantics of the compound is being fulfilled by a manual semantic annotation of the compounds. We will first discuss the guidelines that we used for the annotation process. Apart from a summarisation of the guidelines used, we will also describe the source document and the adaptations we made to it.

After describing the guidelines, we will deal with the annotation process itself. We will present some details about the data we used, how the annotation was performed, as well as some statistics on the agreement between the annotators.

3.1 Scheme and guidelines

Semantic annotation is a very hard task for human annotators. The ubiquitous ambiguity makes it almost impossible to achieve high inter-annotator agreement. It is clear that well-documented guidelines are of the utmost importance.

Our guidelines are based on Ó Séaghdha (2008). When Ó Séaghdha started developing his annotation scheme, there weren't many annotation schemes for compound semantics available. The larger part of the ones that did exist, however, were mere descriptive classifications and did not have explicit guidelines to clarify the scheme. Ó Séaghdha's starting point for the scheme he developed was Judith Levi's 1978 inventory-based model. Six months of annotation trials and scheme improvements led to his current annotation scheme with accompanying guidelines. The main idea is that each compound receives one tag consisting of the broad category in which the compound is semantically situated, the annotation rule that was chosen to arrive at the correct tag and the direction in which this annotation rule is applied. It is not allowed to assign different categories to the same compound.

Although our aim was to stay close to the original annotation guidelines as proposed by Ó Séaghdha (2008), we did make some adaptations to his guidelines other than expanding them with Dutch examples. The main reason for these adaptations was the different setup of our experiment.

The major difference between the two approaches lies in the selection of the compounds to be annotated. We have decided to only deal with regular noun-noun compounds that are not lexicalised (i.e. compounds that cannot be found in the dictionary). The 'regular' aspect of this decision allows us to leave out metaphorical and exocentric compounds from our research. Exocentric compounds have their semantic head outside the compound, which makes them irregular because the compound is not a hyponym of the syntactic head (Plag 2003). They are thus often metaphorical, e.g. *spierbundel* muscle+bundle 'very muscled man'. Compounds that act as proper nouns, or that contain a proper noun, abbreviation, compound, phrase or acronym will also be disregarded, since in many cases their meaning can not be deduced from its parts, notably when the whole compound is a proper noun, e.g. *Leopoldlaan* 'Leopold Avenue'.

The second part of our decision, 'compounds that are not lexicalised', does away with all compounds that can be found in the dictionary, e.g. *voetbal* (football, soccer). Since the goal of our research is to be able to find the meaning of compounds, we do not need to analyse these lexicalised compounds anymore because they already have a dictionary entry that contains the meaning. Luckily, most of the metaphorical and exocentric compounds are already lexicalised, so disregarding them will not influence our coverage too much.

A second reason to not accept lexicalised compounds in our annotation list is the fact that we are designing this experiment to be able to classify newly produced compounds. It will be better to use similar compounds (those of the productive kind and thus not lexicalised) to predict the semantic class of newly produced compounds. Using training and test data from the same frequency level is generally a good heuristic.

Still keeping the research goal in mind (finding the meaning of compounds), knowing the relation between two constituents is not enough. We also have to know the meaning of the separate constituents before the meaning of the entire compound can be found. Our complete compound selection method is thus dependent on a dictionary. The compounds that qualify for annotation are those compounds that are not present in the dictionary but of which the constituents are listed in the dictionary, the exceptions being noted above.

A last adaptation was performed on the examples that accompany the categories. Dutch examples were added to the description of each category. All examples were also provided with the direction of the compound. This information should allow the annotator to get a better understanding of the annotation rules and the direction in which they work.

The full version of the guidelines[1] can be found in the appendix of Verhoeven (2012).

The annotation scheme requires the annotator to assign each compound one out of eleven categories, the rule that the annotator followed to decide on the category and the direction of rule application appropriate for the compound. The eleven categories can be divided in three groups. The first six categories, namely BE, HAVE, IN, ACTOR, INST and ABOUT, are categories that assign a specific semantic class to the compound. The categories REL, LEX and UNKNOWN are used to describe compounds that cannot be classified as one of the other categories, either because the relation between the constituents is unclear (REL, e.g. *Churchilllaan* 'Churchill avenue'); because the compound has a very specific, lexicalised meaning that cannot be brought back to its constituents (LEX, e.g. *loftrompet* praise+trumpet 'praise'); or because the meaning of the entire compound is unclear (UNKNOWN). The last group of categories, MISTAG and NONCOMPOUND, exist for annotation purposes only. They are used to classify words that are present as noun-noun compounds in this list, but are not supposed to be in this list. The MISTAG category is used for words/compounds of which one or both of the constituents is not a common noun. The NONCOMPOUND category refers to sequences that are correctly tagged as regular nouns, but that are not noun-noun compounds for some reason.

The main categories are explained here:

1. The guidelines can also be downloaded from the project website: http://tinyurl.com/aucopro.

- **BE** - This category implies that the compound can be rewritten as 'N2 which is (like) (a) N1' with N1 and N2 being the two constituents of the compound in that order. This includes material-form compounds (e.g. *rubberband* 'rubber tyre') and also most coordinated compounds.

- **HAVE** - All compounds denoting some sort of possession belong in this category. A typical property of this possession is that there should be a one-to-many relationship between the possessor and the possessed. Part-whole compounds (e.g. *autodeur* 'car door'), compounds expressing conditions or properties (e.g. *kankerlijder* 'cancer sufferer', *broodgeur* 'smell of bread') and meronymic compounds (e.g. *groepslid* 'group member') all belong in this category.

- **IN** - Any compound denoting a location in place or time belongs in this category. Examples are: *badkamer* ('bathroom') and *avondspel* ('evening game').

- **ACTOR** - When there is a characteristic event or situation denoted in the compound and one of the constituents is a salient entity, the category is ACTOR. For example, in *huizenbouwer* ('house builder') there is an action of building houses. The *bouwer* refers to a person, which is a salient entity. This compound therefore belongs in the ACTOR category.

- **INST** - This category is the counterpart of the ACTOR category. When the compound denotes a characteristic event or situation and there is no salient entity present, for example because the compound consists of the action itself and the object of this action, the category is INST (referring to 'instrument'). E.g. *smaakbederf* ('flavour decay') where *smaak* is the object of the action *bederf*.

- **ABOUT** - This last semantically specific category deals with topical relations between the constituents of a compound. The typical instantiation of this category is a compound that describes 'an item that is ABOUT something' (Ó Séaghdha 2008, 38). *Geschiedenisboek* ('history book') would be a perfect example for this category.

3.2 Annotation process

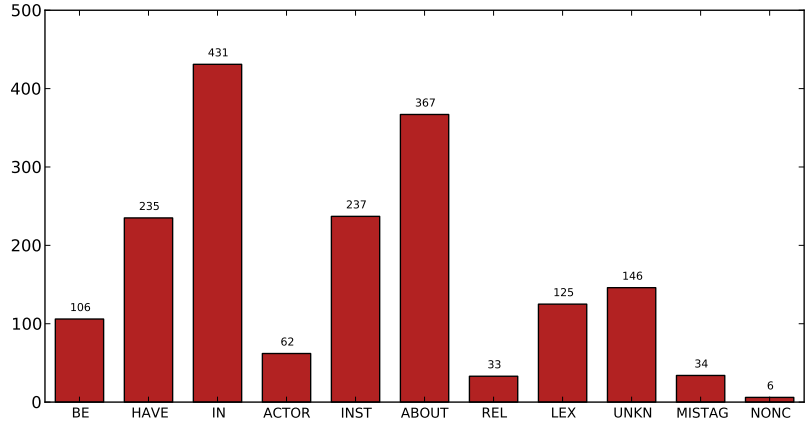
For this annotation task, we used a list of compounds that was extracted from the E-Lex Dutch lexicon[2]. The compounds were already split into constituents and the POS-tags of the constituents were available. Two thousand noun-noun compounds were randomly selected from this list. Those compounds were not allowed to appear in the WNT (Woordenlijst Nederlandse Taal) lexicon but their constituents did have to be present in this Dutch dictionary (Nederlandse Taalunie 2005). Of these 2,000 compounds, 198 double items were removed. Our final compound list for annotation contained 1,802 noun-noun compounds.

The annotation process is also largely inspired by Ó Séaghdha (2008). There were two annotators for this task. The first annotator was a third-year linguistics student at the University of Antwerp. The first author of this article was the second annotator. Both annotators are native speakers of Dutch and have a linguistic background. The first annotator was not involved in the development or adaptation of the guidelines.

The first annotator annotated the entire set of 1,802 compounds. The second annotator annotated 500 compounds so an inter-annotator agreement could be calculated. Half of these 500 compounds were taken from the beginning of the entire compound list; the other half was taken from the end of the compound list. This measure was taken to capture a possible evolution in annotation habits of the first annotator. Figure 1 describes the distribution of the annotation between the classes.

2. This compound list was created by Lieve Macken from the LT3 research group (Language and Translation Technology Team) at Ghent University.

Figure 1: Class Distribution of Annotated Dutch Compounds



3.3 Agreement

The inter-annotator agreement (IAA) of an annotation experiment is a measure of the validity of the manually annotated data. The agreement is a measure of how similar the annotations of different annotators are. The agreement is calculated by dividing the number of equally annotated instances by the total number of instances.

However, this IAA can be a misleading measure when dealing with skewed class distributions. The probability for an instance to be annotated as belonging to a certain class is not equal for each class in this case. The Kappa measure (K) will take the class distributions of the different annotators into account and thus provide a more reliable measure of annotation agreement (Boleda and Evert 2009).

The inter-annotator agreement (IAA) on the categories of the 500 compounds was 60.2% ($K = 0.60$). Although this is somewhat lower than other reported IAA's, e.g. O Seaghdha's IAA was 66.2% with a kappa score of 0.62 (2008), this is not a bad result. We must not forget that semantic annotation is a very difficult task. We do notice that our IAA and Kappa score are very close to each other; this means the two annotators have a very similar category distribution. We also calculated the agreement scores for the broad categories together with the direction. The agreement here is 54.0%. The agreement on the complete annotated information (category, direction and rule) is 46.8%. However, Ó Séaghdha's guidelines were not developed with the intention of maximising the distinctions between rules in the same category (Ó Séaghdha 2008, 45).

There are several factors that are likely to have contributed to our lower IAA. The most important one being that the compounds were not accompanied by their context in our annotation process. This may cause a higher disagreement between the annotators because the context would normally constrain the possible interpretations of a compound. Analysing this confusion matrix (see Table 1) also shows us that there are certain categories that are often disagreed upon by the annotators. Remarkably, this interchangeable aspect works in both directions. The interchanged categories are: IN & HAVE, HAVE & ABOUT, IN & ABOUT, and ABOUT & INST.

This may be an indication that the boundaries between these categories are not sufficiently described in our guidelines. Especially the ABOUT and HAVE category are often interchanged with other categories or each other. Optimising the guidelines by more clearly delineating the boundaries and emphasising the differences between these categories could also raise IAA. It is also possible that the first annotator was not 'skilled' enough in applying the guidelines. Because of the difficulty

		Annotator 1					
		BE	HAVE	IN	ACTOR	INST	ABOUT
Annotator 2	BE	20	3	2	0	3	2
	HAVE	2	40	**16**	1	5	**9**
	IN	2	**9**	87	2	1	**11**
	ACTOR	0	1	0	14	0	2
	INST	2	4	0	1	32	**8**
	ABOUT	4	**7**	**9**	6	**9**	60

Table 1: Confusion matrix of the agreement between the two annotators for the semantically specific classes. The agreements of the interchanged categories are in bold.

of this particular task, it may be necessary to put more time in the training of the annotators and do more test annotations.

The low IAA can also partly be attributed to the non-specific categories. When calculating the IAA solely on the six specific, semantic categories (BE, HAVE, ABOUT, IN, ACTOR, INST), agreement increases to 67.6%. When doing further research, closer attention will have to be paid to the definition and correct use of these less specific categories (LEX, REL, UNKNOWN, MISTAG, NONCOMPOUND).

As a little side experiment, an intra-annotator agreement was also calculated. The same annotator (our student) annotated the first 250 compounds of the list again a month after the first annotation. The agreement (based on the categories) between these two annotations was only 68.2% with a kappa score of 0.68. An overall agreement of 53.5% was achieved. These numbers are of course better than the inter-annotator agreement, but are rather low for a second annotation of the same annotator. This shows again how difficult this task really is, especially when there is no context available.

4. Experiments

The experiments are a variation of those conducted by Ó Séaghdha (2008). We will provide a description of our own experimental setup here.

Our classification experiment is based on a combination of the distributional hypothesis (as proposed above) with the idea of analogical reasoning. It is assumed that the semantic category of a compound can be predicted by comparing compounds with similar meanings (Ó Séaghdha 2008).

When translating the distributional hypothesis from words to compounds, there are different possibilities to be considered. Ó Séaghdha (2008) combines a lexical and relational similarity approach. We have adopted only the lexical similarity approach.

This approach derives a measure of similarity from pairwise similarities between constituents (Ó Séaghdha 2008, 56). In other words, instead of comparing the semantics of the entire compounds, the measure of similarity will be based on the semantic similarities between the constituents of the compounds. The modifiers of the compounds will be compared with each other and the compound heads will be compared with each other. Two compounds that have similar modifying constituents and similar head constituents will be considered as similar on the whole, for example 'flour can' and 'corn bag' will be considered similar because they have similar modifying constituents ('flour' and 'corn' are both types of grain) and similar head constituents ('can' and 'bag' are both types of containers).

4.1 Instance creation

For every compound constituent in our annotated list, the contexts are calculated. The Twente News Corpus (Ordelman et al. 2007), a 340 million word Dutch corpus, was our source of co-occurrence contexts. When the entire corpus has been searched, the lists of context words per constituent are topped off. The 10,000 most frequent context words with their relative frequencies (the number of times the word appeared in the context of the constituent, divided by the frequency of the constituent in the corpus) are stored.

We are, however, not interested in only constructing constituent vectors. For every compound, we create an instance that contains the compound itself, its category, direction and rule (as annotated before), and the relative frequencies for the 1,000 most frequent words for the respective constituents. In total, there are 2,000 features in our vector space. The compound vector is thus a concatenation of the constituent vectors, which is a novel approach for this type of problem. The usual approach is to either add or multiply the constituent vectors together (Mitchell and Lapata 2010). Compounds of which one or both of the constituents did not appear in the corpus, were excluded from our dataset.

Our final datasets only include those compounds that are annotated with a semantically specific category. This means that only compounds with the category tags BE, HAVE, IN, ABOUT, INST or ACTOR will be used for our classification experiments. This leaves 1,447 compounds in our dataset.

The purpose of our research is not merely to be able to classify compounds on the basis of their semantics. We want to investigate in what circumstances this classification works best. A first distinction was made in the compilation of the lists of context words. The assembling was performed in two distinct ways. The first and widely used approach (Schütze 1992, Evert 2010) is to calculate a list of a number (e.g. 10,000) of frequent words in advance and only register the co-occurrences that are present in this list.

In our second approach, the list of context words is calculated after the corpus crawling. For every compound, the 10,000 most frequent words are stored and the list of context words that will be used for the instance creation is calculated by taking those context words that appear in the contexts of most constituents. This approach is thus not based on the absolute frequencies of the words in the corpus. The hypothesis is that this approach might provide us with better results by reducing the data sparsity in the vectors. Since the vectors are designed to contain words that occur in the most contexts, there will be fewer words that have a frequency of 0 in the context of the constituent. Although our 'cont' method theoretically allows for overfitting because the train data is not completely independent from the test data, we verified that there was no noticeable influence on the results. These approaches will respectively be abbreviated as 'freq' (only frequency-based) and 'cont' (occurrence in more contexts) in the results section.

A second variation in our data representation concerns the difference between the morphologically complex forms (or lexical items) and the root forms (or lemmas) of the words in our corpus. In one option, the list of context words contains the lexical items, or tokens, as they appear in the corpus. For example, 'be' and 'is' will be different items in our context list. The other option only allows for the context list to contain lemmas, or root forms, of the words. In this case, 'be' and 'is' will be counted as instances of the same lemma and will fall under 'be' in the context list.

Each approach has its advantages and disadvantages. When using lemmas, there is more room in the 10,000 item list for semantically different items, the morphological and syntax markers of the words that also might provide clues on the semantics of a word are lost. The abbreviations to refer to these approaches are 'lemma' and 'lex'.

Modifying the size of the co-occurrence context of the constituent leads to the third variation in our sets of training data. We investigate how much context of a word is needed to optimally describe its semantics. There will have to be a balance between having a large enough context to describe the constituent's semantics and having a context that is too large and contains words that

no longer have anything to do with the constituent (that are mere noise in the data). Three sizes were chosen for this purpose: a context size of 3, 5 and 10 words in both the left and right context was computed.

All possible variations of the data were combined with each other, resulting in 12 different datasets.

So far, we described a 'bag of words' approach where each token (lemma or lexical item) equals one attribute in the instance vector. Because the compound vector contains 2,000 attributes, this approach is computationally rather expensive and there is reason to try and reduce our vector size for performance sake. One way of achieving this is using principal components analysis (PCA).

PCA is a mathematical transformation of data stored in a matrix. The representation of this data is adapted so that the variance in the data is optimized. The vectors will be reduced in size because correlated variables will be fused. The new attributes of these vectors are called principal components (PCs). The PCs are ordered so that the PC that explains the most variance in the original dataset is the first attribute. The PC with the lowest variance is the last attribute (Smith 2002).

To perform these mathematical transformations on our data, we used the SVD algorithm in the 'PCA Module for Python' as implemented by Risvik (2008). The SVD (singular value decomposition) algorithm is one of the basic algorithms to perform PCA.

New datasets for our experiments were then created using the SVD algorithm. The PCA was performed on the constituent context data. When creating the compound vectors, the first 50 PCs per constituent were selected. Apart from the 'bag of words' data (BOW), we now also have SVD data for every variant. They have 100 attributes per compound (50 per constituent).

For the actual machine learning experiments on the 24 datasets (BOW and PCA, each with 3, 5 or 10 context words, using lemmas or lexical items with our 'cont' and 'freq' method), we used the SMO algorithm, which is WEKA's (Witten et al. 2011) support vector machines (SVM) implementation, and the IB1 algorithm in TiMBL (Daelemans and Van den Bosch 2005). Automatic optimization of the parameters in Weka was performed by the CVParameterSelection function.

We used 10-fold cross-validation; each classifier was trained and tested ten times on a different train and test set. The ten folds cover the whole data set maximally. The average results of these ten runs are a representation of the performance of this classifier.

5. Results

We will first present the results of the SVM machine learning experiment, which will be compared with the TiMBL results. Tendencies that may be present in the results will then be identified and discussed. An error analysis can be found in Section 5.4.

5.1 Main results

To obtain the following results on our classification task for the semantics of Dutch compounds, the machine learning algorithms were provided with the twelve variants of our data (as described in Section 4). The SMO algorithm was used on these twelve variants in their 'bag of words' (BOW) form and in their PCA form. The IB1 algorithm (TiMBL's k-nearest distance algorithm) was used only on the PCA data due to the computational complexity of using TiMBL on high-dimensional datasets.

Since this is the first research on Dutch compound semantics, a baseline of 29.5% will be assumed. This baseline was calculated by dividing the count of the most frequent class (428 instances of class IN) by the total number of compounds in the dataset (1,447). This number represents the accuracy that can be obtained by always guessing IN as the output class. Table 2 presents the micro-average results achieved with the SMO classifier and the results of the IB1 classifier on the PCA datasets.

The results in Table 2 clearly show a significant improvement over the most frequent class baseline. The BOW approach reaches better results, with a maximum of 49.0% F-score, than the PCA

Variants			SMO						IB1		
			BOW			PCA-SVD			PCA-SVD		
			Prec	Rec	F-Score	Prec	Rec	F-Score	Prec	Rec	F-score
Freq	Lemma	3	47.7	47.5	47.6	44.5	48.1	44.6	44.2	44.4	44.2
Freq	Lex	3	47.6	48.0	47.8	41.7	46.2	41.7	45.6	45.5	45.4
Cont	Lemma	3	**49.5**	**48.8**	**49.0**	45.2	47.9	45.1	44.9	45.1	44.9
Cont	Lex	3	46.7	47.0	46.8	43.7	46.8	42.9	45.2	45.3	45.2
Freq	Lemma	5	46.6	46.6	46.5	45.4	48.2	44.2	45.6	45.5	45.5
Freq	Lex	5	47.7	48.0	47.8	43.0	47.6	43.6	46.0	45.8	45.8
Cont	Lemma	5	45.7	45.5	45.5	**45.8**	**48.4**	**45.2**	45.4	45.6	45.5
Cont	Lex	5	47.8	48.4	48.0	44.4	48.4	43.9	45.1	45.3	45.1
Freq	Lemma	10	47.0	47.0	46.9	45.5	47.9	42.9	44.3	44.6	44.4
Freq	Lex	10	47.2	47.7	47.4	44.2	48.4	42.5	45.1	45.7	45.3
Cont	Lemma	10	46.4	46.3	46.3	44.2	47.9	42.8	44.5	44.5	44.4
Cont	Lex	10	47.4	48.0	47.6	42.3	47.8	41.8	45.6	45.9	45.7

Table 2: Main Results on BOW and PCA Instances using 10-fold Cross-Validation. The best variant for each system is marked in bold.

approach, with a highest F-score of 45.2%. The F-scores for the BOW approach vary from 45.5% to 49.0%, which gives an average F-score of 47.2%. This average shows that the BOW approach seems to do better than the PCA approach, where an average F-score of 43.4% was achieved with results ranging from 41.7% to 45.2%. Although the PCA approach with the SVD algorithm reaches significant results, it is outperformed by the BOW approach. It is however the difference in macro-average F-score (PCA 36.4% vs. BOW 43.9%), and not micro-average F-score, that is statistically significant ($p = 0.012 < 0.05$). This is mainly because there is a high difference in macro-average recall ($p = 0.0019 < 0.05$) between the two approaches. This implies that a BOW approach has a positive effect on the recall of the smaller categories. When calculating statistical significance for the micro-average F-scores, the BOW approach does not show a statistically significant improvement ($p = 0.373 > 0.05$) over the PCA approach, although the micro-average precision of the BOW approach is quite a bit higher than the precision of the PCA approach. Significance was calculated using approximate randomization testing.

The F-scores achieved by the IB1 algorithm on the PCA data range from 44.2% to 45.8%. The average F-score of all the variants is 45.1%.

The best results were achieved by the BOW approach. It seems that some of the information in the instances is lost during the PCA calculation. Nevertheless, the results from the PCA data are also significantly better than the random baseline.

Additional Experiment using 8 Categories

The entire research is based on the classification of our data for six semantically specific classes. It may be interesting to investigate the accuracy of a system that also takes the less specific REL and LEX categories into account. A classification was performed on the data set with 8 categories with the same specifications as our best performing PCA data set: Cont Lemma 5.

Table 3 shows a drop of 4.2% in overall F-score. The REL and LEX categories achieve a low accuracy of 13.6% and 19.5% F-score. This is an indication that the REL and LEX categories are indeed much less specific than the other six categories and are therefore less learnable by our context-based classifier. Including these two categories also seems to bring down the accuracy of the other categories, which is especially noticeable in the IN category.

	6 Cat	8 Cat
BE	10.9	13.7
HAVE	29.9	29.1
IN	62.3	57.4
ACTOR	27.6	30.3
INST	32.0	31.6
ABOUT	55.8	55.7
REL	-	19.5
LEX	-	13.6
Weighted Avg.	45.2	41.0

Table 3: Comparison of F-Scores per Class with 6 or 8 Categories.

5.2 Tendencies

Table 4 is included in this section to illustrate some of the tendencies that can be noticed in the main experimental results. It contains the averages of the results shown in Table 2.

	Avg. F-score BOW SMO	Avg. F-score PCA SMO
3	**47.8**	43.5
5	47.0	**44.2**
10	47.1	42.5
Freq	47.5	43.2
Cont	47.2	43.6
Lemma	46.9	**44.1**
Lex	**47.5**	42.7

Table 4: Average SMO F-scores for Different Experimental Aspects. The best variant for each system is marked in bold.

A first observation of this table teaches us that there is hardly any difference in results due to the manner of context calculation. The hypothesis that 'more context' would perhaps raise the results could not be proven.

The number of context words that are being taken into account does seem to have an influence on the results, though the different approaches do not all show the same outcome. The average SMO BOW results show a better performance of the classifier when using three context words. The average SMO PCA results show an improvement of the F-scores when using five context words. Finally, the results also show an influence of the type of corpus elements used. The SMO BOW results show better results when using lexical items instead of lemmas. The SMO PCA results point in the other direction. This SMO PCA approach was, however, the one with the lowest F-scores. The other results thus have higher credibility.

5.3 Result analysis

In this section, a result analysis of the classification of the best performing experiment is presented. The idea is not to identify tendencies across different approaches but to have a more detailed look at the results of the SMO classifier on the data set that yields the best results.

		Classifier					
		INST	HAVE	ABOUT	IN	ACTOR	BE
Annotation	INST	**0.41**	0.13	0.11	0.09	0.05	0.16
	HAVE	0.09	**0.34**	0.13	0.12	0.10	0.19
	ABOUT	0.25	0.20	**0.57**	0.09	0.16	0.11
	IN	0.16	0.22	0.12	**0.62**	0.24	0.24
	ACTOR	0.03	0.02	0.03	0.02	**0.44**	0.04
	BE	0.05	0.09	0.05	0.06	0.02	**0.26**

Table 5: Confusion Matrix of the Classification with the Best Results. The values are the column probabilities.

Figure 2: Comparison of Class Distributions between Annotation and Best Classifier

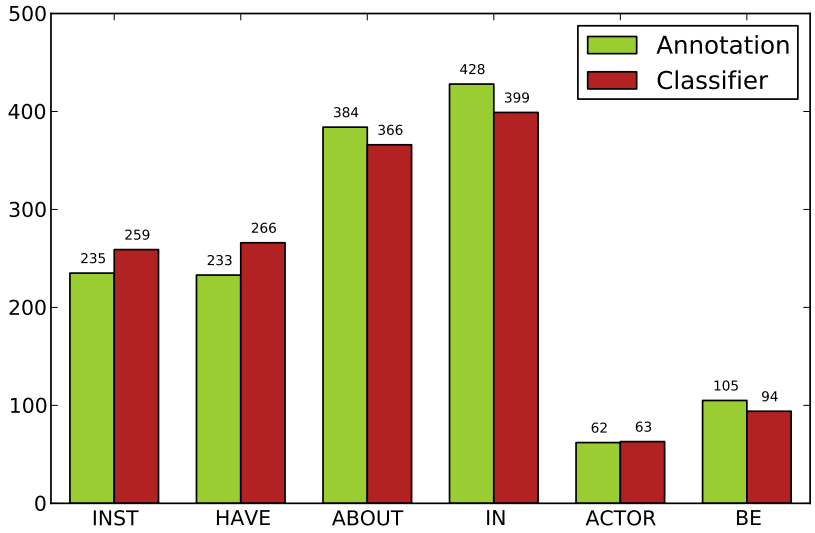


Table 5 presents the confusion matrix of the best classifier; this is the SMO algorithm on the BOW Cont Lemma 3 data set. This table contains the column probabilities, which represent the probability of the classifier assigning the column class to the gold label represented by the row. The sum of each column is thus one. We notice many misclassifications, which is normal with an F-score of 49.0%, but there seem to be no structural errors in this confusion matrix.

Figure 2 shows us that the class distributions of the classifier are very similar to the class distributions of the annotation. This aspect was apparently learned well by the SMO algorithm. Table 6 provides us with the results by class that this classifier achieves. As expected, the classes with higher frequencies reach a higher accuracy, which makes sense since there is more training information available on this class. The BE class has a rather low frequency and has the lowest accuracy with an F-score of 24.1%. Interestingly, the ACTOR category, which has the lowest frequency, does have the third highest accuracy (44.8%), suggesting that it is a very easy class to learn.

	Precision	Recall	F-score
INST	40.9	45.1	42.9
HAVE	33.8	38.6	36.1
ABOUT	57.4	54.7	56.0
IN	62.2	57.9	60.0
ACTOR	44.4	45.2	44.8
BE	25.5	22.9	24.1
Weighted Avg.	49.5	48.8	49.0

Table 6: Accuracies by Class of the Confusion Matrix in Table 5

5.4 Error analysis

In this section, we take a detailed look at the classification of the instances of the best performing class (IN) of the best variant of our experiment: SMO BOW Cont Lemma 3.

According to the annotation, 25 out of 45 compounds in this class were correctly classified. This makes 20 compounds that were misclassified. Of these 25, there are however 5 compounds that may be incorrectly annotated. For example, the compounds *ovendeur* 'oven door' and *pistoolheft* 'pistol grip' are annotated as IN, where they would be better annotated as HAVE (part-whole).

Of the 20 misclassified compounds, only 3 are truly incorrect. In 4 cases, both the annotation and the classification seem appropriate. These are context-dependent matters such as *badkuur* bath+treatment 'spa treatment', which may be classified as IN (treatment in a bath) or as INST (bath serves as participant in the treatment). There are also 5 cases where both annotation and classification appear to be wrong and even 8 cases where the annotation seems incorrect and the classification indicates the right relation. Examples of both annotation and classification going wrong include *katoog* 'cat eye', which was classified as BE but is supposed to be HAVE and *galakoets* 'gala carriage' which was classified BE but is actually ABOUT. Some examples of the classification being correct and the annotation being wrong, are *koorlessenaar* 'choir desk' (correctly classified as HAVE) and *ovulatiestoornis* 'ovulation disturbance' (correctly classified as INST).

These occurrences are of some concern. They mostly show our annotation is still far from perfect and the annotators will need more guidance. There are also indications, namely the 8 misannotated but correctly classified compounds, that the classifier actually works rather well.

6. Conclusion

This paper focused on the semantic analysis of Dutch noun-noun compounds by using computational classification methods. The noun-noun compounds were semantically annotated in advance. These annotations were performed using guidelines that describe different semantic categories of compounds. The semantic analysis by the classifier is based on distributional information about the constituents of the compound, i.e. information about the words that appear in the context of these constituents in a corpus.

1,802 compounds were annotated, of which 1,447 belong to one of six semantically specific classes. The overlap between annotators was 60.2% (with K=0.60) which already compares favourably with previous annotations for English using the same annotation scheme, although we believe that a better training of the annotators and using the compounds in their context will lead to even better annotations.

Our supervised machine learning experiments were novel in the sense that they were performed on Dutch compounds, which has never been done before, and because we used a new approach to combining the constituent information into one compound vector, namely simple concatenation of the constituent vectors. Our experiments reached significant results. Our best performing experiment

had a micro-average F-score of 49.0%, which is significantly higher than the most frequent class baseline. As a first result, this already compares favourably with the 58.8% F-score (accuracy of 61.0%) reached on English compounds using the same method (Ó Séaghdha 2008). The BOW approach appears to consistently outperform the PCA approach.

The variants of the experiment with 3 and 5 context words on both sides of the constituent, performed better than those with 10 context words. This is probably because the context loses its specificity when it is that large. It no longer only describes the constituent, but also includes too much information on irrelevant words. There is no real difference in performance between the 'cont' and 'freq' measure. There is a difference between the performance of the 'lex' and 'lemma' variants, but it is not clear-cut. The SMO BOW experiments showed better performance using the lexical items, the SMO PCA experiments preferred using the lemmas. However, these are mere tendencies that can be seen in the experiment averages. These are not visible in individual experiments.

The results of our experiments using the PCA approach turned out not only to be a significant improvement over the majority baseline, but also rather close to the results of the BOW approach. Only on the recall of the smaller categories is the BOW approach significantly better. However, this difference makes the entire F-score of the PCA approach noticeably lower than that of the BOW approach. These results are promising because they might allow us to create smaller datasets, which would speed up our experimentation process. Smaller data sets are easier to handle by machine learning algorithms. The lower results indicate a possible loss of information in the calculation of the PCs.

Future research will focus on the optimisation of the experiment in order to achieve better results that would compare more favourably to the state-of-the-art in experiments for English noun compounds by taking some of the following remarks into account.

During the annotation process, it will be necessary to better educate the annotator about the guidelines before the annotation starts. Probably some more adaptation to the guidelines is also appropriate so as to be able to better distinguish between the categories that showed a lower agreement. The annotation of the compounds in context (with example sentences) is also recommended in future research. Considering a compound in context would be a more natural language usage and would constrain the possible interpretations of a compound. These measures should raise the agreement, and hopefully also the performance of the classifier.

As for the experimental setup, the 10 context words variant will probably not perform any better than using 3 or 5 context words. Crawling a corpus and storing 10 context words left and right for every constituent is also computationally rather expensive, which makes us even more inclined to discard this approach. It may be useful to introduce more variance in the lower range of context words, e.g. also do experiments with 1, 2 or 4 context words left and right to the constituent in the corpus.

It will no longer be necessary to distinguish between the 'freq' and 'cont' approach. They perform practically equally good, but the 'freq' approach is a lot easier and faster. A choice is readily made. The distinction between 'lex' and 'lemma' in performance may need some more attention, but it does not seem as if one of the two will outperform the other.

There are still other factors to our research that may be interesting to investigate. Changes in the number of most frequent words might have an influence on our system's performance. Also the kind of tokens we use in this most frequent 'words' list can be of importance. Apart from the lexical items and lemmas, special attention could be given to the effect of taking into account only function words or only content words.

Acknowledgements

This research was funded by a joint research grant of the Nederlandse Taalunie (Dutch Language Union) and the Department of Arts and Culture (DAC) of South Africa for a project on automatic

compound processing (AuCoPro[3]). This is a mutual project by research groups of the North-West University (South Africa), the University of Antwerp (Belgium) and Tilburg University (The Netherlands).

References

Barker, Ken and Stan Szpakowicz (1998), Semi-automatic recognition of noun-modifier relationships, *Proceedings of the 17th International Conference on Computational Linguistics*, Morgan Kaufmann, San Francisco, pp. 96–102.

Boleda, Gemma and Stefan Evert (2009), Computational lexical semantics: Inter-annotator agreement, *European Summer School in Logic, Language and Information (ESSLI-09)*, Association for Logic, Language and Information (FoLLI).

Booij, Geert (2002), *The Morphology of Dutch*, Oxford University Press, Oxford, UK.

Daelemans, Walter and Antal Van den Bosch (2005), *Memory-Based Language Processing*, Cambridge University Press, Cambridge, UK.

Evert, Stefan (2010), Distributional semantic models, *Proceedings of Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, Los Angeles.

Finin, Timothy W. (1980), The semantic interpretation of compound nominals, *Urbana*.

Girju, Roxana, Dan Moldovan, Marta Tatu, and Daniel Antohe (2005), On the semantics of noun compounds, *Computer Speech and Language* **19**, pp. 479–496.

Harris, Zellig (1968), *Mathematical Structures of Language*, Interscience, New York.

Hinrichs, Erhard, Verena Henrich, and Reinhild Barkley (2012), Using part-whole relations for automatic deduction of compound-internal relations in GermaNet, *Language Resources and Evaluation*, Springer.

Kim, Su Nam and Timothy Baldwin (2005), Interpreting semantic relations in noun compounds via verb semantics, *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, Association for Computational Linguistics, Sidney, pp. 491–498.

Lapata, Mirella and Frank Keller (2004), The web as a baseline: Evaluating the performance of unsupervised web-based models for a range of NLP tasks, *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, Boston, pp. 121–128.

Lauer, Mark (1995), *Designing Statistical Language Learners: Experiments on noun compounds*, PhD thesis, Macquarie University.

Miller, George A. (1995), WordNet: A lexical database for English, *Communications of the ACM* **38** (11), pp. 39–41.

Mitchell, Jeff and Mirella Lapata (2010), Composition in distributional models of semantics, *Cognitive Science* **34** (8), pp. 1388–1429.

3. http://tinyurl.com/aucopro

Moldovan, Dan, Adriana Badulescu, Marta Tatu, Daniel Antohe, and Roxana Girju (2004), Models for the semantic classification of noun compounds, *Proceedings of the HLT-NAACL Workshop on Computational Lexical Semantics*, Association for Computational Linguistics, Boston, pp. 60–67.

Nakov, Preslav (2008), Noun compound interpretation using paraphrasing verbs: Feasibility study, *Proceedings of the 13th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA'08)*.

Nastase, Vivi, Jelber Sayyad-Shirabad, Marina Sokolova, and Stan Szpakowicz (2006), Learning noun-modifier semantic relations with corpus-based and WordNet-based features, *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*, American Association for Artificial Intelligence, Boston, pp. 781–787.

Nederlandse Taalunie (2005), Bronbestand woordenlijst Nederlandse taal. http://www.inl.nl/tst-centrale/nl/producten.

Ó Séaghdha, Diarmuid (2007), Annotating and learning compound noun semantics, *Proceedings of the ACL 2007 Student Research Workshop*, Association for Computational Linguistics, Prague, pp. 73–78.

Ó Séaghdha, Diarmuid (2008), *Learning compound noun semantics*, PhD thesis, University of Cambridge, Cambridge, UK.

Ó Séaghdha, Diarmuid (2009), Semantic classification with WordNet kernels, *Proceedings of NAACL-HLT 2009: Short Papers*, Association for Computational Linguistics, Boulder, Colorado, pp. 237–240.

Ó Séaghdha, Diarmuid and Ann Copestake (2007), Co-occurrence contexts for noun compound interpretation, *Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, Association for Computational Linguistics, Prague, pp. 57–64.

Ó Séaghdha, Diarmuid and Ann Copestake (2008), Semantic classification with distributional kernels, *Proceedings of the 22nd International Conference on Computational Linguistics (COLING-08)*, Association for Computational Linguistics, Manchester, pp. 649–656.

Ordelman, Roeland, Franciska de Jong, Arjan van Hessen, and Hendri Hondorp (2007), TwNC: a multifaceted Dutch news corpus, *ELRA Newsletter* **12**, pp. 3–4.

Plag, Ingo (2003), *Word-Formation in English*, Cambridge University Press, Cambridge, UK.

Risvik, Henning (2008), PCA module for Python. http://folk.uio.no/pca_module.

Rosario, Barbara and Marti A. Hearst (2001), Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy, *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 82–90.

Schütze, Hinrich (1992), Dimensions of meaning, *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, IEEE Computer Society Press, Los Alamitos, CA, pp. 787–796.

Smith, Lindsay I. (2002), A tutorial on principal components analysis. http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf.

Tratz, Stephen and Eduard Hovy (2010), A taxonomy, dataset, and classifier for automatic noun compound interpretation, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Uppsala, pp. 678–687.

Verhoeven, Ben (2012), *A computational semantic analysis of noun compounds in Dutch*, Master's thesis, University of Antwerp, Antwerp, Belgium.

Verhoeven, Ben and Gerhard B. van Huyssteen (2013), More than only noun-noun compounds: Towards an annotation scheme for the semantic modelling of other noun compound types, *Proceedings of the 9th Joint ISO - ACL SIGSEM Workshop on Interoperable Semantic Annotation (ISA-9)*.

Verhoeven, Ben, Walter Daelemans, and Gerhard B. van Huyssteen (2012), Classification of noun-noun compound semantics in Dutch and Afrikaans, *Proceedings of the Twenty-Third Annual Symposium of the Pattern Recognition Association of South Africa (PRASA 2012)*, Pretoria, South Africa, pp. 121–125.

Wijaya, Derry Tanti and Philip Gianfortoni (2011), "Nut Case: What does it mean?": Understanding semantic relationship between nouns in noun compounds through paraphrasing and ranking the paraphrases, *Proceedings tof the 1st International Workshop on Search and Mining Entity-Relationship Data (SMER-11)*, Glasgow.

Witten, Ian H., Eibe Frank, and Mark Hall (2011), *Data Mining: Practical Machine Learning Tools and Techniques (Third Edition)*, Morgan Kaufmann, Burlington, MA.