# A Computational Semantic Analysis of Noun Compounds in Dutch

Ben Verhoeven

| | |
|---|---|
| Promotor : | Prof. Dr. Walter Daelemans |
| Assessor: | Dr. Guy De Pauw |

Universiteit Antwerpen

Academiejaar 2011 - 2012

Ondergetekende, Ben Verhoeven, student Taalkunde, verklaart dat deze scriptie volledig oorspronkelijk is en uitsluitend door hemzelf geschreven. Bij alle informatie en ideeën ontleend aan andere bronnen, heeft ondergetekende expliciet en in detail verwezen naar de vindplaatsen.

The undersigned, Ben Verhoeven, student in Linguistics, hereby declares that this thesis is entirely original and exclusively written by himself. When using and borrowing information and ideas of others, explicit and detailed references to the original sources are in place.

Antwerpen, 13 augustus 2012

Verhoeven

This thesis was submitted at the University of Antwerp in August 2012.
Since then some minor adaptations were made.
Final version: November 2012.

# ABSTRACT

This thesis describes the first attempt to semantically analyse Dutch noun compounds using the distributional hypothesis. The automatic analysis of compound semantics has its uses in machine translation, information extraction, information retrieval and question answering systems. Using Ó Séaghdha (2008) as a source of inspiration, a list of 1802 noun compounds was constructed and annotated. The annotators had an annotation scheme and guidelines available. This scheme described six specific semantic categories (BE, HAVE, IN, ACTOR, INST, ABOUT) and five categories for les 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 considered a classification task for which we can use machine learning algorithms. Context information on the constituents of the compounds was used to create instance vectors for the classifier to train on. In certain variants of the experiment, principal component analysis (PCA) was used as a means of reducing the vector's number of dimensions. Implementations of 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) vectors using the SVM algorithm. The PCA vectors 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. Furthermore, the BOW approach significantly outperforms the PCA approach on the recall of the smaller categories. The distributional hypothesis, having already proven its value in English research on compound noun semantics, turns out to also work well on Dutch compounds. Further research to improve our initial results is desirable.

Keywords:
NOUN COMPOUND, SEMANTICS, DUTCH, NATURAL LANGUAGE PROCESSING, COMPUTATIONAL LINGUISTICS

# TABLE OF CONTENTS

# 0. PREFACE

This thesis is submitted in partial fulfilment of the requirements for the Master of Arts in Linguistics with a specialisation in Computational Psycholinguistics at the University of Antwerp, Belgium.

Writing a thesis is the task of an individual student, but often, some help is vital to successfully complete it. The author of this thesis had the luck of being surrounded by many helpful people who contributed to this thesis by providing reading material, giving feedback, explaining some new topics or supporting in any other way. I would like to use this preface to express my gratitude for all this help.

First and foremost, Prof. Dr. Walter Daelemans (CLiPS – University of Antwerp) has been an enormous support as my thesis supervisor. Thank you for the suggestions, the corrections and the guidance throughout this year.

Second, there are several people that helped out with certain practical aspects. I want to thank Dr. Lieve Macken from the LT3 Research Group (Language and Translation Technology Team) at University College Ghent for providing a list of split compounds that was extracted from the e-Lex Corpus. Many thanks to Jana Declercq, student in Literature and Linguistics at the University of Antwerp, for her efforts in semantically annotating the list of noun compounds used in our experiments. I could not have performed all my computational experiments without the computer power that I was able to use at the CTexT research group (Centre for Text Technology) at the North-West University in Potchefstroom, South Africa.

I want to express my appreciation for the explanations and documentation giving to me by Dr. Vincent Van Asch (CLiPS) on the calculation and interpretation of statistical significance, and Dirk Snyman (CTexT), who also proofread parts of my thesis, on the use of the WEKA software.

Lastly, I have had the opportunity of doing an internship at the CTexT research group in Potchefstroom, South Africa as part of a project on Automatic Compound Processing. This was an enormous learning opportunity for which I am very grateful. I want to thank Prof Dr. Gerhard Van Huyssteen, who also provided feedback on the annotation guidelines, and Prof. Dr. Walter Daelemans for the confidence they have in me. I acknowledge the support of the Nederlandse Taalunie (Dutch Language Union) and the Department of Arts and Culture (DAC) of South Africa for the joint grant that made this project, and thus also my internship, possible.

# 1. INTRODUCTION

## 1.1. Contextualisation

Whether a computer will ever fully understand natural language is a question that does not yet have a conclusive answer. Some researchers believe this will never happen (e.g. Winograd & Flores, 1986; Salaberry, 1996). Others are more optimistic and proclaim that it will become possible when natural language processing (NLP) technology and methods have sufficiently advanced (e.g. Ogden & Bernick, 1997; Wolfram, 2010). The issue of computational understanding of natural language remains the topic of quite some debate in the research field of NLP. If computers are ever to really understand natural language, there are still many problems we will have to deal with. Solving any of these problems brings us closer to a better system at every turn.

One of the problems that a computer faces in trying to understand natural language is the productivity that a language exhibits in using newly created words.

The ability of a language to constantly produce new words is practically endless. The processes responsible for this word formation are derivation and compounding. Derivation does not pose much of a problem since many derivations of words are already present in the lexicon as known words and new 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 almost the same meaning. The small variations in meaning are due to the shift to a different part of speech. Compounding, however, is not an easy problem to deal with.

There are several reasons why compounds form a greater problem in the computational understanding of natural language. The four reasons below were identified by Girju et al. (2005).

- The meaning of a compound is a combination of the meaning of its constituents and the semantic relations between these constituents are only implicitly present.
- The meaning of compounds can be idiosyncratic. For example, in order to understand *UN meeting,* you need to know that 'UN' stands for 'United Nations'.
- Sometimes, more than one semantic relation can be identified between the constituents of the compound.
- The interpretation of compounds can be highly context-dependent. For example, *chair city* can have different meanings in different contexts. It might mean that it's a city where a lot of chairs are produced. It can also mean that this city is the chair of some kind of board or council of cities.

## 1.2. Research Goals

The intention of this thesis is to develop a first-generation automatic semantic classifier for Dutch compounds. The specific research goals can be summed up as follows:

- The creation of a semantically annotated list of compounds.
- Exploratory research on the feature creation for the classification of Dutch compounds and presenting initial performance results.
- The investigation of the performance of the experiments using measures of dimensionality reduction, like Principal Components Analysis (PCA)[1].

## 1.3. Applications

Section 1.1 stated that natural language understanding would benefit from the semantic analysis of compounds. There are some specific applications in natural language processing that are worth discussing in this section because they are directly influenced for the better by an improved compound understanding.

An obvious application that would benefit from automatic compound analysis is machine translation (MT). Since not every language uses compounds as productively as English or Dutch, 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 so easily be translated to the French 'auberge à Anvers'.

Information extraction (IE), information retrieval (IR) and question answering (QA) systems can also be improved by better compound understanding. The system needs to know which compounds and (para)phrases co-refer to be able to gather the necessary data on the considered topic (Nakov, 2008).

## 1.4. Structure

In Chapter 2, the theoretical linguistic background of compounds is discussed. This chapter focuses on the problem of defining the notion of compounding and it presents different semantic considerations on the topic.

[1] You will find an introduction to PCA in section 5.2.3.

Chapter 3 provides an overview of the related research on automatic processing of compound semantics. This chapter has been divided in sections on the annotation process and guidelines and sections on the methods of compound comparison.

Chapter 4 is a description of the annotation guidelines and process. The adaptation of the annotation guidelines that were adopted from Ó Séaghdha (2008) is here presented, together with statistics on how our annotators performed on this task. The complete and detailed annotation guidelines can be found in the Appendix.

In Chapter 5 we present our own experimental setup. First the important theoretical assumptions that lie at the heart of our research, such as analogical learning and the distributional hypothesis, are presented. These are followed by an explanation of the vector creation for our compound classification experiment. Following these practical notions are some sections on the machine learning aspect of the experiment. The algorithms used are presented, as well as the statistical measures of evaluating the outcome of the experiment.

Chapter 6 provides and analyses the results of the different variants of our experiments.

Finally, Chapter 7 discusses these results and attempts to formulate some conclusions on compound processing in Dutch. A number of suggestions for further research conclude this thesis.

# 2. Linguistic Description of the Domain

The following chapter will provide some linguistic background on compounding. The focus will be on noun compounds, but this will be integrated in a general overview of compounding. We will first discuss the definition of compounding and some issues surrounding this definition. Morphology and grammar of compounds will also be addressed, as well as the linguistic research on semantic aspects of compounds. These properties of compounding will be treated in general and will also be applied specifically to Dutch.

## 2.1. Definition

Over the years, several linguists have formulated a definition for the process of compounding. A consensus is unfortunately nowhere in sight. Intuitively, one would say that a compound is a new word formed by joining two or more words together. Our intuition is here supported by Katamba (1994 in Benczes, 2006), Bauer (2009) and Plag (2003), among others. However, defining the notion of a compound is not as easy as it seems. In English, this definition already provides some problems. The claim that a compound is a new word can be disputed since English does not always morphologically connect compounds. Although the constituents are joined in a syntactic structure, the surface form does not show one new word. Spaces can be left between the joined words.

A word thus has to be understood as 'lexeme' in this definition (Bauer, 2009). We will elaborate on this alternative definition below.

The problem with finding a definition of compounding exists on two levels. On the micro level, we should ask ourselves whether the constituent elements of a compound are free-standing words or rather stems or roots. Second, it is sometimes problematic to distinguish compounds from phrases or derived words. This problem is situated on the macro level (Lieber & Štekauer, 2009).

### 2.1.1. Micro-Level Problem

The micro problem is partially solved by Bauer (2009) by changing 'word' in the definition to 'lexeme'. A lexeme is an abstract unit of language that is closely connected to the notion of 'lemma'. Lexemes are all morphemes that have a semantic interpretation of their own, even if they never occur alone (SIL International, 2003). This nuance in terminology was needed to include e.g. neoclassical compounds, where lexemes of Greek or Latin are combined with another lexeme to form new combinations that were not present in the original language (Plag, 2003).  Some English examples of neoclassical compounds are: ‘*hydro*power’, ‘*retro*-design’ or ‘*bibliography’*, but they also exist in Dutch (e.g. ‘*democratie’*, ‘*biologie’*, ‘*elektro*motor’) and other languages (Lüdeling, 2009). These Greek or Latin morphemes do not occur on their own and are therefore not to be considered words. This type of lexemes can also be found in compounds of the language in question. An example for Dutch is: *oer*tijd (prehistory), where ‘oer’ means something like ‘primal’ and ‘tijd’ is ‘time’.

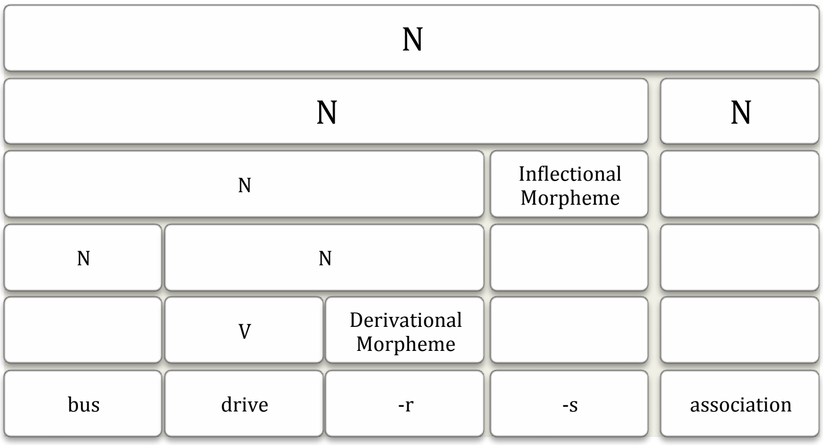
The micro problem is with this only partially solved. There are still other elements that can be constituents of compounds that aren’t really words. A compound like ‘pipe-and-slipper husband’ takes a phrase as left-hand constituent. If we take this kind of compounds into account, our final definition of compounding should be that ‘a compound is a word that consists of two elements, the first of which is either a root, a word or a phrase, the second of which is either a root or a word’ (Plag, 2003:135). Note that ‘words’ are still part of the definition. Roots or stems do not explain the compounds with inflected constituents, e.g. ‘parks commissioner’ or ‘woman’s magazine’. The compound as a whole is uninflected, but the left-hand member is here plural or genitive (Plag, 2003; Lieber & Štekauer, 2009).

The abovementioned definition considers a compound as consisting of two elements. Of course there are compounds that have more constituent elements. It is however ‘generally possible to analyse polymorphemic words as hierarchical structures involving binary (i.e. two-member) subelements’ (Plag, 2003:133). In other words, derivations, inflections and compounds (words with more than one morpheme) are always constructed on top of each other. They do not affect the base element at the same time. We can represent these analyses in bracketed notations or tree structures (Plag, 2003). Here is an example of the notations for the analysis of the binary structure of words.

e.g. bus drivers association

a. [ [ [ bus [ drive + r ] ] s ] association ]

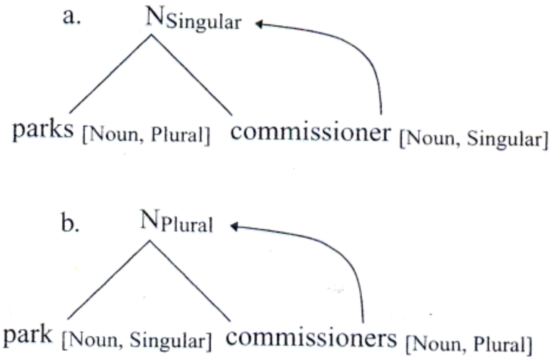
b.



When analysing multi-word compounds, they will all have a similar binary structure. Since a compound is considered a word and the constituent elements of compounds can naturally be words, it is safe to state in our definition that each compound combines two elements.

Where compounding and derivation both have this binary structure, only compounding has a property that allows the repeated creation of the same kind of structure. We can say that the compound formation rules are recursive, whereas derivation is generally not as recursive as compounding. A derivation can be made of a derivation, e.g. ((friend$_N$ + ly)$_{Adj}$ + ness)$_N$ or in Dutch ((twijfel$_N$ + achtig)$_{Adj}$ + heid)$_N$ 'dubiousness', but this is always finite. Only 2 or 3 derivations can be made, whereas the process of compounding is theoretically infinite. Note that the longer and more complex a compound becomes, it will be harder for the language user to produce or understand it. We will therefore seldom come across compounds with more than 5 constituents.

The most common interpretation of a compound in English (and Dutch) is that it has the left-hand member modifying the right-hand member. There is a modifier-head structure present in these compounds wherein the head is grammatically and semantically the most important unit (Plag, 2003; Lieber & Štekauer, 2009). Semantically, the entities denoted by the compound are a subset of the entities denoted by the head. For example, 'bookshelves' are a subset of 'shelves' and a 'bar manager' is a type of manager. Grammatically, the compound inherits its grammatical features from the head. This is called feature percolation. If we take the 'parks commissioner' example again, Plag (2003:136) shows us schematically the inheritance of grammatical features from the head. Here we have an example of the inheritance of the plural inflection, but other features are of course possible, e.g. gender inflection in languages that make these distinctions.



In English, it happens to be that the modifier-head structure almost always has the head after the modifier. Because this appeared to be an important feature of English compounding, Williams (1981:248 in Plag, 2003; Lieber & Štekauer, 2009) formulated this as the Right-Hand Head Rule.

The modifier-head structure is not the only possible structure. A compound can also have coordinated elements. In section 2.3.1.3 we will provide you with some more details on these coordinated compounds.

### 2.1.2. Macro-Level Problem

The macro problem with compounding is situated on the syntactic level and is far less solved than the micro problem. Sometimes it is hard to distinguish between phrases and compounds (Lieber & Štekauer, 2009). The problem is especially present with noun compounds. Some academics consider noun compounds altogether as noun phrases with a head noun preceded by a modifying noun that assumes the function of an adjective. However, a distinction has to be made because it is not only possible to mistake a compound for a phrase, it is also possible that one mistakes a phrase for a compound (Plag, 2003).

The inseparability criterion is one of the best criteria to distinguish compounds from phrases in English. We will demonstrate this with an example from Lieber & Štekauer (2009:11). 'While it is possible to insert another word into the phrase *a black bird*, e.g. *black ugly bird*, no such insertion is permitted with the compound *blackbird*.' This criterion works very well, although there are still exceptions to this rule (Coolen, 1994; Lieber & Štekauer, 2009).

Another alleged way to make the distinction between phrases and compounds in English (and again, also in Dutch) is phonological. The stress in pronunciation is supposed to be on the left-hand element in a compound and on the right-hand element in a phrase. This compound stress rule, as opposed to the nuclear stress rule in phrases that places the stress on the last word of the phrase, was formalised by Chomsky and Halle in 1968 (Plag, 2003). There are however a lot of exceptions to this rule. Giegerich (2009) even claims the rule to be a myth. We will not elaborate on this subject, since we're mostly interested in the written version of compounds.

More information on compound stress can be found in the following publications: Plag (2006), Plag (2003), Lieber & Štekauer (2009), Don (2009) for Dutch, and Giegerich (2009).

### 2.1.3. Compounding in the World's Languages

The word formation process of compounding has received very little attention in linguistic typology so far (Guevara & Scalise, 2009). It is however very present in the world's languages. Some scholars even suggest compounding to be a language universal (Bauer, 2009). There is some evidence for this hypothesis, for example the widespread presence of compounds in pidgin languages. There are, however, also several cases of languages whose grammar does not mention compounding. The definition of compounding also plays a large role in this respect because some languages only have disputed forms of compounding (e.g. noun incorporation, which is the combination of a noun modifier and a verb head where "the verb head can behave just like any other finite verb form" (Spencer, 2005:89)) that - depending on the definition - can be seen as compounds (Bauer, 2009).

In the Germanic languages, compounding is the major means of vocabulary expansion. Other languages (e.g. Turkana) use compounds only in the formation of names. There is evidently a huge variation in the use and productivity of compounding. The trouble is that this variation is seemingly random. There does not appear to be evidence for any correlation with other elements in the language structures (possibly because for many languages we do not have qualitative descriptions of word formation processes), therefore, the construction of a typology on compounding remains impossible (Bauer, 2009).

### 2.2. More on Morphology: Linking Letters

In the sections above, we have already discussed some morphological aspects of compounding. Yet, there is still a morphology-related topic that we would like to deal with: linking letters. This is an important trait of Dutch compounding that cannot be ignored.

Linking letters - or linking elements, in general - between the compound's constituents occur in many languages. These linking letters are typically semantically empty, but often have the genitive (and sometimes the plural) inflectional morpheme as its etymological source (Booij, 2010; Bauer, 2009; Don, 2009). Linking letters are most prevalent in Germanic languages (except for English), but also occur in other language families (Krott et al., 2007).

Linking letters with genitive origin:

- 'her-en-huis' (Dutch for 'mansion', lit. 'lord's house')
- 'koning-s-kroon' (Dutch for 'king's crown')

(Booij, 1996)

- 'koken-s doar' (Frisian for 'kitchen door')

(Booij, 2010)

### 2.3. Semantics

In this section on compound semantics, we will discuss two ways of looking at compound semantics. We will give them different names for clarity's sake. The first approach takes the status of the head of the compound as the defining factor to classify a compound in the different semantic classes (Scalise & Bisetto, 2009). In the second approach, semantic models are developed to describe and classify the semantic relation between the compound constituents.

### 2.3.1. Semantic Classes

The following section will provide you with an overview of the prevalent semantic classes that are present in the linguistic literature on compounds (e.g. Coolen, 1994; Plag, 2003; Scalise & Bisetto, 2009). However, not all scholars accept theses classes. Some have small variations on these classes, some have proposed new classes and others do not recognise all these classes. Still, there is a large consensus on three main classes with a number of subclasses. For a complete overview and comparison of possible classification methods of compounds, we refer to Scalise & Bisetto's chapter 'The Classification of Compounds' in The Oxford Handbook of Compounding (2009).

As mentioned before, when talking about the modifier-head structure, these semantic classes are based on the status of the head of the compound. In section 2.1.2, we remarked that the notion of 'head' can have a semantic and a syntactic interpretation. Both interpretations will be discussed below.

#### 2.3.1.1. Endocentric Compounds

The endocentric compound has its semantic head within the compound itself. This means that the compound 'denotes a subclass of the referents of the head' (Plag, 2003:145). In other words, '[t]he compound as a whole is a hyponym of the head element' (Coolen, 1994:6). These endocentric compounds are mostly noun compounds and are the most common compounds in English and Dutch. For example, 'boekenrek' (Dutch) or 'book shelf' is a kind of shelf that is used to hold books.

As for a syntactic interpretation of endocentricity, a compound can be said to be syntactically endocentric when it has its syntactic head within the compound itself and the compound 'inherits most of its […] syntactic information from the head' (Plag, 2003:135).

According to the right-hand head rule (see above), the right-hand member of the compound will almost always be the syntactic and semantic head of the compound in most Germanic languages.

#### 2.3.1.2. Exocentric Compounds

The exocentric compounds are a small class of compounds. In contrast to the endocentric compounds, the compounds in this class have a semantic head outside the compound (Plag, 2003). For example, a ‘must-have’ is not some sort of ‘have’, nor is a ‘lion heart’, in its usual metaphorical meaning, a subtype of the entity ‘heart’. Exocentric compounds have such specific meanings that most of them are lexicalised. This class is obviously less productive than the endocentric class. Exocentric compounds also occur more with adjectives than with nouns.

Another term that is used quite often for this class of compounds is *bahuvrihi*. This term comes from the Sanskrit grammar tradition and originally means ‘(having) much rice’ (Scalise & Bisetto, 2009). It currently applies to all exocentric compounds, but traditionally refers to what is presently called possessive compounds. Possessive compounds are exocentric compounds that ‘denote an entity that is characterized (sometimes metaphorically) by the property expressed by the compound’ (Plag, 2003:146). ‘Redskin’ and ‘roodhuid’ (Dutch) are great examples. They refer to an entity, which is a person here, that has a red skin, namely a Native American.

While an endocentric compound is practically always both syntactically and semantically endocentric, this is not the case with exocentric compounds. A compound can be syntactically and semantically exocentric, but can also be semantically exocentric and syntactically endocentric. An example of the former can be ‘must-have’, which is a noun as a whole but has a verb as its syntactic head. There is also no semantic head present in this compound. An example of the latter is: ‘redskin’ which is not a subset of the noun ‘skin’ but we observe that it does adopt some syntactic features of the head, namely the part of speech (Plag, 2003).

#### 2.3.1.3. Copulative Compounds

A third main class of compounds is the copulative class. This class is also called *dvandva* in the Sanskrit tradition, which means ‘couple’. This refers of course to the fact that in copulative compounds there is no semantically more prominent constituent. Both members of the compound equally contribute to the interpretation and meaning of the compound (Plag, 2003). Compounds like ‘bittersweet’, ‘woman doctor’ or ‘Austria-Hungary’ are examples of this class (Scalise & Bisetto, 2009).

Again, the true meaning of the Sanskrit term has been twisted a bit. The original *dvandva* compounds are those that ‘associate two individual elements without reference to any of them as a separate entity’ (Scalise & Bisetto, 2009:36). Examples[2] of these true *dvandva* compounds are ‘mātāpitarau’ (Sanskrit: mother and father), ‘yamakawa’ (Japanese: mountains and rivers) and ‘maxeropiruno’ (Modern Greek: fork and knife). Notice that these compounds all denote a sort of duo (a real ‘couple’), which is a type

[2] Examples from: http://en.wikipedia.org/wiki/Dvandva [06/04/2012]

of compound that – to my knowledge – does not occur in Western-European languages. The current meaning of *dvandva* has lost this 'duo' connotation, but still uses two heads to express the meaning of the compound.

There are however two other subtypes of copulative compounds. These subtypes are frequently used in academic literature but are not generally accepted. The terms for the subtypes and the entire class of copulative compounds are used interchangeably (Coolen, 1994; Scalise & Bisetto, 2009).

An appositional compound is a copulative compound that 'refers to one entity that is characterized by both members of the compound' (Plag, 2003:146). The compound is both an entity of constituent one and an entity of the second constituent. A 'bastard brother' or 'bastaardbroer' is both a brother and an illegitimate child. Other examples include 'actor-director' and the Dutch word 'bombrief' ('bomb letter').

The coordinative compound, according to Plag (2003), is a copulative compound that is usually part of a larger compound where the *dvandva* denotes 'two entities that stand in a particular relationship with regard to the following noun' (Plag, 2003:146). The 'mother-daughter bond' is a bond between mother and daughter. The 'north-south stream' is a stream that runs from the north to the south.

### 2.3.2. Semantic Models

Semantic models are designed to describe and classify the semantic relations between the compound constituents.

When looking for a manner to investigate compound semantics, one has a choice of three viewpoints. All proposals of semantic models can be grouped into one of the three types. Ó Séaghdha (2008) made a clear overview that discusses these types of semantic models. We will provide a summary here. This section will be focusing on noun compounds only since only they have received research attention over the years. The semantics of other sorts of compounds has barely been investigated.

#### 2.3.2.1. Inventory-Based Theories

The early approaches in semantic modelling of noun compounds focused on description. Linguists adopting this inventory-style approach documented 'the variety of semantic relations observed in attested compounds' (Ó Séaghdha, 2008:17). These semantic relations were generalised to a restricted set of relations. These relations were seen as the retrieval of the full semantic structure of the noun phrase at a deeper representational level.

Some criticism has come from different angles:

- The variety of compound relations is so great that listing them is impossible. This criticism especially relates to metonymic and metaphoric compounds.
- The proposed relations are too general and vague. This problem is called ‘analytic indeterminacy’. Some compounds can have multiple analyses.
- The restricted inventories give too impoverished a representation of compound semantics. On this view, the meaning of a compound cannot be reduced to one of a small number of general relations.

(Ó Séaghdha, 2008:18-19)

### 2.3.2.2. Pro-Verb Theories

The authors of the pro-verb theories are the ones that formulated the latter criticism on inventory-based semantic models. They believe that dealing with the semantics of a compound by simple describing the relation between the constituents is a strong reduction of the actual meaning of the compound. Instead of generalising the compound meaning, pro-verb models will underspecify the representation of this meaning and shift the task of the further interpretation of the compound to pragmatics or world knowledge. ‘The semantics of a compound is then simply the assertion of an unspecified relation between the referents of its constituents’ (Ó Séaghdha, 2008:19).

Despite its linguistic value, it is immediately obvious that a pro-verb approach will be useless in current computational linguistics.

### 2.3.2.3. Integrational Theories

The integrational theories originate in the tradition of cognitive linguistics. It is believed that there is no divide between compositional semantic structures and pragmatic kinds of conceptual and contextual knowledge. Integrated representations of compounds are generated by combining aspects of the constituent nouns. So-called *event frames* are a central kind of knowledge in this process. These are ‘schematic representations of the events or situations in which an entity typically plays a role’ (Ó Séaghdha, 2008:20). The integrational theory claims that, in order to arrive at the compound meaning, one has to place the constituents in an event frame where they belong together. For example, when interpreting the compound ‘boekenrek’ (‘bookshelf’) the language user will combine his knowledge of books and shelves to arrive at an event frame where the shelf can be used to hold the books.

Pro-verb and integrational theories do not recognize the restricted set of classes as they are proposed in the inventory-based models. They do, however, recognise that there are regularities and patterns in the way language users experience and conceptualise their environment. It is assumed that there are certain abstract templates in creating compounds. For example, locative relations ('library table') or part-whole relations ('car wheel') are often present in compounds because these relations provide more information about the entity.

Strangely enough, the prevalent templates are very comparable to the inventory-style rules.

#### 2.3.2.4. Example of an Inventory-Based Model

Despite the criticism, the inventory-based model is used most nowadays by computational linguists for the automated semantic analysis of compounds. An early example of such a model, that was however not yet developed for computational purposes, was constructed by Judith Levi in 1987 and included nine Recoverably Deletable Predicates (RDPs). The name of the RDPs refers to the problem that appears when the full semantic structure of the compound cannot be recreated from the surface structure, which is a case of *irrecoverable deletion*. It is assumed that the deleted predicate of the compound constituents can in fact be recovered (Ó Séaghdha, 2008).

The semantic classes with examples, as proposed by Levi (Ó Séaghdha, 2008:17):

| | | | |
|---|---|---|---|
| CAUSE$_1$ | *flu virus* | CAUSE$_2$ | *snow blindness* |
| HAVE$_1$ | *college town* | HAVE$_2$ | *company assets* |
| MAKE$_1$ | *honey bee* | MAKE$_2$ | *daisy chains* |
| USE | *water wheel* | | |
| BE | *chocolate bar* | | |
| IN | *mountain lodge* | | |
| FOR | *headache pills* | | |
| FROM | *bacon grease* | | |
| ABOUT | *adventure story* | | |

These classes can be used for analysis as follows. An *adventure story* is a *story* ABOUT *adventure*; a *mountain lodge* is a *lodge* ON a *mountain* (IN = location). 'The three RDPs CAUSE, HAVE and MAKE each have two variants, as either the head or modifier of a compound can fill the first argument of these predicates, while the other arguments are either symmetric (BE) or restricted to taking the compound head as first argument' (Ó Séaghdha, 2008:18).

Other inventory-based models use, for example, insertion of the constituents in thematic role slots of generalised verbs (Lees, 1970) or paraphrasing prepositions (Butnariu et al., 2010).
Unfortunately, we were not able to find any semantic models specifically for Dutch compounds. Most of the current research seems to be focusing on English.

In our computational consideration of compound meaning, the inventory-based theories will be our starting point. As will be discussed in Chapter 4, Ó Séaghdha (2008) based his annotation scheme for the semantic analysis for compounds on the above inventory-based scheme by Levi.

Before the annotation scheme and process will be discussed, an overview will be presented of the past related research on computational compound semantics.

# 3. RELATED RESEARCH

When overviewing the past research on compound semantics, some tendencies can easily be observed. First, as far as we are aware, all research so far has focused on noun compounds. Other sorts of compounds, such as verb-noun or adjective-verb compounds have been disregarded. Second, most research only considers two-word noun compounds. The exceptions on this tendency are the research of Girju et al. (2005) who also perform experiments with three-word noun compounds, and the research of Girju et al. (2007) and other scholars that participated in the SemEval-2007 Task 4, which dealt with classifying relations between nominals. This task thus exceeded the compound category.

There are also two main points of divergence in past research. Many variations on the classification schemes have been proposed. Some of the newly proposed schemes are similar to existing schemes, others differ fundamentally.

The second discrepancy in the related research has to do with the information that is used by the computational system to classify the compounds in the dataset. This chapter will discuss the different choices researchers have made on these two subjects.

## 3.1. Classification Schemes for Computational Research on Compound Semantics

In Chapter 2, we have already mentioned the theoretical attempts to compound classification such as the inventory-based scheme developed by Levi (1979 in Ó Séaghdha, 2008). The classification schemes for computational research in compound semantics all seem to follow this inventory-based tradition. Early birds in the computational research are Warren (1978 in Rosario & Hearst, 2001), Finin (1980) and Lauer (1995). This research will come up again when discussing the different kinds of classification schemes.

Two types of schemes can be distinguished. There are those that base their semantic classes on prepositions only, and those that base their semantic classes on a predicate. We will further divide the latter type in subtypes with predicate-based definitions as classes and with true paraphrasing predicates as classes.

### 3.1.1. Preposition-based Semantic Classes

Preposition-based semantic classes are an abstract way of classifying constituent relations in compounds. A compound is classified as 'FOR' when one can paraphrase the compound as *constituent*

*FOR constituent*. A *balletzaal* ('ballet hall'*)* can thus be a 'zaal voor ballet' ('hall for ballet'), which implies some sort of purpose.

Lauer (1995) was one of the first to propose a preposition-based classification scheme. Apart from his preposition categories (of, for, in, at, on, from, with about), he also had categories for copula compounds (both modifier and head classify the object) and verbal-nexus compounds (the modifier is subject or object of the nominalised verb head) (Lauer, 1995:161).

Lauer's classification scheme has been used and tested by Girju et al. (2005) and Lapata & Keller (2004).

A problem with this approach arises when the same preposition can be used in different contexts, with different meanings. The Dutch *rivierbrug* ('river bridge') can be paraphrased as 'brug OVER rivier' ('bridge over river'), but *avonturenboek* ('adventure book') can be paraphrased as 'boek OVER avonturen' ('book about adventures'). The same principle holds for English, e.g. the preposition 'of' can denote possession, but it can also denote the topic of something. Actually, a preposition-based classification scheme should take this polysemy into account and disambiguate between the different meanings of the prepositions, for example by creating two classes for 'of' OF-possession and OF-topic. This would reduce the abstractness of the preposition-based schemes.

The preposition-based classification is nowadays considered too abstract for the analysis of compound semantics. Most researchers will use predicate-based classification schemes that are better suited for the coverage of deeper semantic relations.

### 3.1.2. Predicate-based Semantic Classes

The predicate-based approach can be regarded as having two variants. The first one provides predicates for the compound by classifying the compounds according to semantic definitions. These definitions describe predicates for the compounds. The second variant will provide actual paraphrases for the compound by inserting some verbal element between the constituents.

#### 3.1.2.1. Relations Described by Definitions and Classes

This approach is actually very similar to the one described by Levi in her classification scheme (1979 in Ó Séaghdha, 2008). Many others have followed her example and have either borrowed her classification for their experiments (Nakov, 2008) or have created their own classification scheme in the same tradition.

Ó Séaghdha designed a classification scheme with 11 categories based on Levi's scheme (Ó Séaghdha, 2007; Ó Séaghdha, 2008; Ó Séaghdha & Copestake, 2007). An inter-annotator agreement kappa score was reported of 0.62. This classification scheme has been adopted for our research as well. The complete (slightly adapted) guidelines for this scheme can be found in the Appendix. An overview of this scheme can be found in Chapter 4.

Moldovan et al. (2004) proposed a more specific scheme with 35 semantic relations. This scheme has also been used by Girju et al. (2005; 2007). Girju et al. compared the preposition-based classification with the predicate-based semantic classification. They found the predicate-based method to consistently outperform the preposition-based method. However, the annotation seems to be easier using preposition-based classes, but this is probably due to the fewer classes of the preposition-based approach. Girju et al. (2005) report an inter-annotator agreement kappa score of 0.80 for Lauer's 8 prepositional classes (1995) versus a kappa score of 0.58 for 35 semantic relations proposed by Moldovan et al. (2004).

Tratz & Hovy (2010) use a taxonomy of 43 relations of their own making that they describe as being similar to the scheme designed by Warren (1978 in Tratz & Hovy, 2010). They report a rather poor inter-annotator agreement of 52.3% but they possess the largest dataset of annotated compounds with over 17,500 instances.

All above schemes are open-domain schemes. There are however researchers that have posited domain-specific classification schemes. Rosario & Hearst (2001) describe a classification scheme that is specifically designed for use in biomedical texts.

#### 3.1.2.2. Paraphrasing

Instead of creating rules that describe the predicate of the constituents of a compound, researchers have recently paid rather a lot of attention to compound paraphrasing as a way to create a predicate for the compound constituents.

In initial research, Kim & Baldwin (2005) used the scheme designed by Barker & Szpakowicz (1998). This 20-member classification scheme paraphrases the implicit relation with a verb and/or a preposition. They later expanded this scheme with the concept of 'seed verbs'. These seed verbs are considered the hypernyms of actual verbs that are mapped onto them (Kim & Baldwin, 2006). The reported agreement between annotators on their annotation task was 52.3%.

Wijaya & Gianfortoni (2011) describe topics that are very similar to the seeds verbs described by Kim & Baldwin (2006).

## 3.2. Methods of Compound Comparison: Feature Selection

Throughout the years, several methods have been proposed to construct features for the semantic classification of compounds. Most of these methods are either based on a corpus, or on a semantic taxonomy. These methods will be thoroughly discussed and we will conclude this section with an introduction to some alternative methods that have recently been developed.

### 3.2.2. Corpus-based Methods

Corpus-based methods are those methods for vector creation that rely on large corpora to extract the features from. It is assumed that the context in which the constituents of the compound occur provide information on the semantics of this constituent. Thus, words with similar contexts in a corpus will have similar meanings. Three methods will be discussed here. Lexical similarity and relational similarity are both methods that use proximity characteristics of the compounds. Using proximity features implies the use of words that are in the neighbourhood of the considered word in the corpus as features to include our vector. The third method uses grammatical information that is extracted from the corpus.

#### 3.2.2.1. Proximity Features

Lexical similarity

Lexical similarity, also called attributional similarity (Turney, 2006), is a measure for comparing the context of the compound. The hypothesis is that “compounds with semantically similar constituents will encode similar relations” (Ó Séaghdha & Copestake, 2007:57). The context-based semantics of the modifiers of the considered compounds will be compared with each other, and the head constituents will be compared with each other. The similarities between both constituents will be combined to calculate a measure of similarity for the entire compounds.

Ó Séaghdha has used corpus methods in much of his research, alone or with Ann Copestake. They report the following results with lexical similarity: using kernel methods as machine learner yielded an accuracy of 54.95% (Ó Séaghdha, 2007; Ó Séaghdha & Copestake, 2007) and later an accuracy of 61% has been reached (Ó Séaghdha, 2008; Ó Séaghdha & Copestake, 2008).

### Relational similarity

In the relational similarity approach, “two pairs [of constituents] are assumed 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). For example, when looking for the relational similarity between ‘car door’ and ‘flower pot’, the relational similarity will be calculated on the words in contexts where the constituents ‘car’ and ‘door’ occur together and likewise for ‘flower’ and ‘pot’.

This method has been used by Ó Séaghdha & Copestake (2007) with an accuracy of 42.34%. Ó Séaghdha (2008) reports an accuracy of 52.6%. In both studies kernel methods were used to achieve these results.

Tratz & Hovy (2010) used the relational similarity approach together with WordNet and morphological features. Their results will be discussed in section 3.2.2.

Lapata & Keller (2004) are the only ones reporting on an unsupervised method for the analysis of compound semantics while using the web as a corpus to compute their relational similarity. This similarity measure is based on web counts for phrases *Noun P Noun,* where P belongs to a predefined set of prepositions. They achieve an accuracy of 55.71% on their unsupervised web-based similarity. This contrasts with their unsupervised corpus-based accuracy of 27.85%. Despite these good results for unsupervised models, they report that web-based models still fail to outperform recent supervised models and are rather a good baseline than an alternative to these recent classifiers.

### Combination

In some research, these proximity measures were also combined to achieve a higher accuracy.

Turney (2006) reports an F-score of 56.5% using Latent Relational Analysis. Ó Séaghdha & Copestake’s (2007) lexical similarity results improve with about 2%, reaching 56.55% combined accuracy. Ó Séaghdha (2008) reaches 62.7% combined accuracy. This indicates that relational and lexical similarities at least partly encode different information on the compound semantics. They provide different ‘views’ on the same semantic relation (Ó Séaghdha & Copestake, 2007). This information is complementary to each other, thus significantly improving the model’s performance.

#### 3.2.2.2. Grammatical Collocations

Instead of simply taking the nearby words of the compound constituent of an information source, Nastase et al. (2006) use only grammatical collocations of this constituent. This collocation includes words that appear with the target word in a grammatical relation, e.g. subject, object, etc.

The results of Nastase et al.'s experiment (2006) will be discussed in section 3.2.2 because they also used WordNet information to boost their classifier's performance.

### 3.2.3. Taxonomy-based Methods

These methods, also called semantic network similarity (Ó Séaghdha, 2009), base their features on a word's location in a taxonomy or hierarchy of terms.

#### 3.2.3.1. WordNet Similarity

WordNet (Miller, 1995) is probably the most well known semantic network (it is also one of the few) and is therefore used most. Some researchers have done experiments using only WordNet features. Others have combined WordNet features with corpus-based or other features.

These WordNet features are usually a vector of hypernyms (words whose meaning include the source word, a generalisation) of the source word. These hypernyms can describe relations such as 'is a', 'has part' and 'is made of'.

Kim & Baldwin (2005) describe an experiment where they calculated the word similarity of the head nouns of the compounds to be compared and multiplied this with the word similarity of the modifier nouns. The new compound will be assumed to belong to the same semantic class as the compound with the highest comparison score. No real machine learning is in order here. Their experiment yielded an accuracy of 53.3%.

The following researchers combine or compare corpus-based methods and WordNet-based methods.
Nastase et al. (2006) report an F-score of 82.47% on a combination of their grammatical collocations and WordNet hyponym information.

Ó Séaghdha (2007) uses binary feature vectors "whereby a vector entry is 1 if the item belongs to or is hyponym of the synset corresponding to that feature, and 0 otherwise" (Ó Séaghdha, 2007:77). An accuracy of 58.35% is reported.

Tratz & Hovy (2010) do not only use corpus methods and WordNet features, they also include morphological features (e.g. the last 3 letters of the constituent). These features are used as Boolean values. They report an accuracy of 79.3%. The influence of the WordNet sense gloss words on the model's performance is stressed.

#### 3.2.3.2. Wikipedia-based

An experiment using Wikipedia as a source for measuring word similarity has been described by Strube & Ponzetto (2006 in Ó Séaghdha & Copestake, 2007). Wikipedia as a semantic network can be an interesting tool because it is "more explicit in its description of relations between real-world entities than typical text corpora" (Ó Séaghdha & Copestake, 2007:64).

#### 3.2.3.3. Other Lexical Hierarchies

Rosario & Hearst (2001) describe an experiment using a domain-specific (biomedical) lexical hierarchy as information source for their classifier. The features are thus the locations where the term is positioned in the MeSH (Medical Subject Headings) lexical hierarchy. An accuracy of about 60% was reported.

### 3.2.4. Other Methods

Girju et al. (2005) used the WordNet approach to assemble their vectors, but they added word sense disambiguation (WSD) to one of their variants to be able to make a comparison. Their best results were obtained using the SVM machine learning algorithm. Without WSD an accuracy of 72.59% was achieved. The WSD made the accuracy rise to 83.93%. These results prove the usefulness of word sense disambiguation in the compound semantics problem.

Finally, there are also some researchers who took the possible paraphrases of a compound into account to arrive at a well performing classifier. Nakov (2008), Kim & Baldwin (2006) and Wijaya & Gianfortoni (2011) gathered paraphrases of compounds using Amazon Mechanical Turk. Where Nakov (2008) uses only these paraphrases to train his classifier, both Kim & Baldwin (2006) and Wijaya & Gianfortoni (2011) first generalise these paraphrases to a hypernym. This is called the 'seed verb' (Kim & Baldwin, 2006) or the 'topic' (Wijaya & Gianfortoni, 2011). Nakov (2008) reports an accuracy of 78.4%, while Kim & Baldwin (2006) achieved only a 52.6% performance.

Nakov (2008) goes one step further than the other two researches. He reports a significant correlation (37.3%) between the verbs found by web searches of the compound constituents and the annotated paraphrases. This means we will also be able to classify unseen compounds by using web search to find paraphrases.

## 3.3. Summary

This chapter provided an overview of the past research in the semantic analysis of compounds. We have only found research on compounds in English, most of them dealt only with two-word noun compounds.

We have first discussed different ways of using classification schemes while considering the reported inter-annotator agreements. Then, we compared different information sources that researchers use to train their classifiers on. Overall, it seems that models that use more than one information source perform better than others. Not only when using different variants of the same method – e.g. lexical vs. relational similarity (Ó Séaghdha, 2008) – but also when combining completely different approaches – e.g. using both corpus-methods and WordNet similarity (Tratz & Hovy, 2010). Different methods provide different information on the compounds, so combining them will enhance a model's performance.

# 4. Annotation: Guidelines, Motivation and Process

The current chapter 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.

In the first section, we will discuss the guidelines, or protocol if you will, that we used for the annotation process. Apart from a summarisation of the guidelines used, we will also consider the source document and the adaptations we made to it.

The following section 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.

## 4.1. Annotation Scheme and Guidelines

### 4.1.1. Source

The annotation guidelines that fit very well into the goal of our annotation, namely the description of the semantics of Dutch compounds, were created by Ó Séaghdha (2008). We already discussed his work in the Related Research section, so we will not explain his entire research again. There are however some particular aspects that are worth taking a closer look at.

In his research, Ó Séaghdha strives to achieve state-of-the-art performance on the task of automatically assigning semantic categories to English compounds. For this, he used the fresh perspective of only using annotated compounds with their extracted context from corpus data as input to his machine learning algorithms. Previous research had mainly focused on using lexical databases with semantic taxonomies such as WordNet for this purpose (Ó Séaghdha, 2008).

Semantic annotation is a very hard task for human annotators. The ubiquitous ambiguity makes it almost impossible to achieve high inter-annotator agreement (an accuracy measure that is discussed in section 4.2.3). According to Ó Séaghdha (2008:28), there are 5 criteria to keep in mind to successfully create an annotation scheme:

- Coverage: The inventory of informative categories should account for as much data as possible.
- Coherence: The category boundaries should be clear and categories should describe a coherent concept.
- Generalisation: The concepts underlying the categories should generalise to other linguistic phenomena.
- Annotation Guidelines: There should be detailed annotation guidelines, which make the annotation process as simple as possible.
- Utility: The categories should provide useful semantic information.

Every criterion should be maximally considered while creating an annotation scheme. Of course, sometimes there is interference between these criteria. If you want to enlarge the coverage of the scheme, your categories will probably be more specific and your guidelines will probably be more difficult. The scheme creator should try to find a balance between these criteria that fits their annotation task (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 (see above, section 2.3.2.4.). Six months of annotation trials and scheme improvements led to his current annotation scheme with accompanying guidelines. The scheme allows for an annotator to describe the semantic relation between the two constituents of a noun-noun compound. 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. You will find the annotation scheme below. More explanation on the guidelines can be found in section 4.1.3. We adopted Ó Séaghdha's annotation scheme but did make some adaptations to the guidelines. Those changes are described in the following section 4.1.2.

| Relation | Rule | Definition | Example |
|---|---|---|---|
| BE | 2.1.1.1.<br>2.1.1.2.<br>2.1.1.3. | Identity<br>Substance – Form<br>Similarity | *guide dog*<br>*rubber wheel*<br>*cat burglar* |
| HAVE | 2.1.2.1.<br>2.1.2.2.<br>2.1.2.3.<br>2.1.2.4.<br>2.1.2.5. | Possession<br>Condition – Experiencer<br>Property – Object<br>Part – Whole<br>Group – Member | *family firm*<br>*coma victim*<br>*sentence structure*<br>*computer clock*<br>*star cluster* |
| IN | 2.1.3.1.<br>2.1.3.2.<br>2.1.3.3.<br>2.1.3.4. | Spatially Located Object<br>Spatially Located Event<br>Temporally Located Object<br>Temporally Located Event | *pig pen*<br>*air disaster*<br>*evening edition*<br>*dawn attack* |
| ACTOR | 2.1.4.1.<br>2.1.4.2. | Sentient Participant – Event<br>Participant – Participant (more prominent is sentient | *army coup*<br>*project organiser* |
| INST | 2.1.5.1.<br>2.1.5.2. | Non-Sentient Participant – Event<br>Participant – Participant (more prominent is not sentient | *cereal cultivation*<br>*foot imprint* |
| ABOUT | 2.1.6.1.<br>2.1.6.2.<br>2.1.6.3.<br>2.1.6.4. | Topic – Object<br>Topic – Collection<br>Focus – Mental Activity<br>Commodity – Charge | *history book*<br>*waterways museum*<br>*embryo research*<br>*house price* |
| REL | 2.1.7.1 | Other Non-Lexicalised Relation | *fashion essentials* |
| LEX | 2.1.8.1. | Lexicalised Compound | *life assurance* |
| UNKNOWN | 2.1.9.1. | The Meaning is Unclear | *similarity crystal* |
| MISTAG | 2.2.1.1. | Incorrectly Tagged | *legalise casino* |
| NONCOMPOUND | 2.2.2.1. | Not a 2-Noun Compound | *[ hot water ] bottle* |

(Ó Séaghdha, 2008:34)

### 4.1.2.Adaptation

Although our aim was to stay close to the original annotation guidelines as composed 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. We will now provide you with an overview of the changes in the guidelines.

The major difference between our 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. Compounds that act as proper nouns, or that contain a proper noun, abbreviation, phrase or acronym will also be disregarded.

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 gloss that contains the meaning. Luckily, most of the metaphorical and exocentric compounds are already lexicalised, so disregarding them will not influence our coverage of the different compounds 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 in the categories. 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. You also have to know the meaning of the separate constituents before you can figure out the meaning of the entire compound. 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.

All these exceptions are dealt with in rule 1.4 of the guidelines in the appendix. Rule 1.4 thus comprises the adaptation of former rules 1.4 and 1.5 (Ó Séaghdha, 2008). Most of these exceptions will receive the REL-tag should they occur in the annotation list.

The former rules 1.6 and 1.7 (Ó Séaghdha, 2008) of the guidelines discussed the treatment of characteristic situations or events. Since these rules deal with the same topic, they were combined into the current rule 1.5.

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 piece of information should allow the annotator to get a better understanding of the annotation rules and the direction in which they work.

#### 4.1.3. Description

The current section will summarise the annotation scheme and guidelines. The full version of these guidelines can be found in the Appendix. The annotation scheme was adopted from Ó Séaghdha (2008).

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* in which the rule is 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. The respective explanations for this is either because the relation between the constituents is unclear; because the compound has a very specific, lexicalised meaning that cannot be brought back to its constituents; or because the meaning of the entire compound is unclear. The last group of categories, MISTAG and NONCOMPOUND, is 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.

Some more explanation is in order for the reader to get an idea of the meaning of these specific semantic categories.

- **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. Other more special uses of this category can be found in the guidelines.

## 4.2. Annotation Process

### 4.2.1. Data

For this annotation task, we used a list of compounds that was extracted from the E-Lex Dutch lexicon[3]. 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, 2011). Of these 2000 compounds, 198 double items were removed. Our final compound list for annotation contained 1802 noun-noun compounds.

Our annotation does not yet follow rule 1.2 of the guidelines. This rule states that a compound should be semantically interpreted in context during annotation. Due to a combination of time constraints and

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

corpus unavailability, we were unable to provide example sentences of the considered compounds at the time of annotation. We do support the idea that considering a compound in context would be a more natural language usage. It also has a high chance of improving the accuracy of annotation because the context usually constrains the possible ambiguities of a word. Further research with new annotations will definitely be conducted in accordance with this rule.

As a solution, we asked our annotators to annotate the most typical meaning of the compound. This was a necessary measure, but it produces some difficulties (and lower annotation accuracy than a compound in context) since different annotators may have a different idea about the most typical meaning of a compound.

### 4.2.2. Process

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. This student was hired to work on this annotation task. The author of this thesis 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. She was first introduced to the guidelines by the present author. The guidelines were explained, as well as the goal of the research. Then they both annotated 50 different compounds that were not in the official compound list. These were compared and corrected together. During the annotation process, some more feedback was sent by e-mail but this was kept to a minimum so as not to compromise the agreement results.

The first annotator annotated the entire set of 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. We will expand on the inter-annotator agreement in section 4.2.3.

The first annotator completed the annotation of 1802 compounds. Figure 1 describes the distribution of the annotation between the classes.

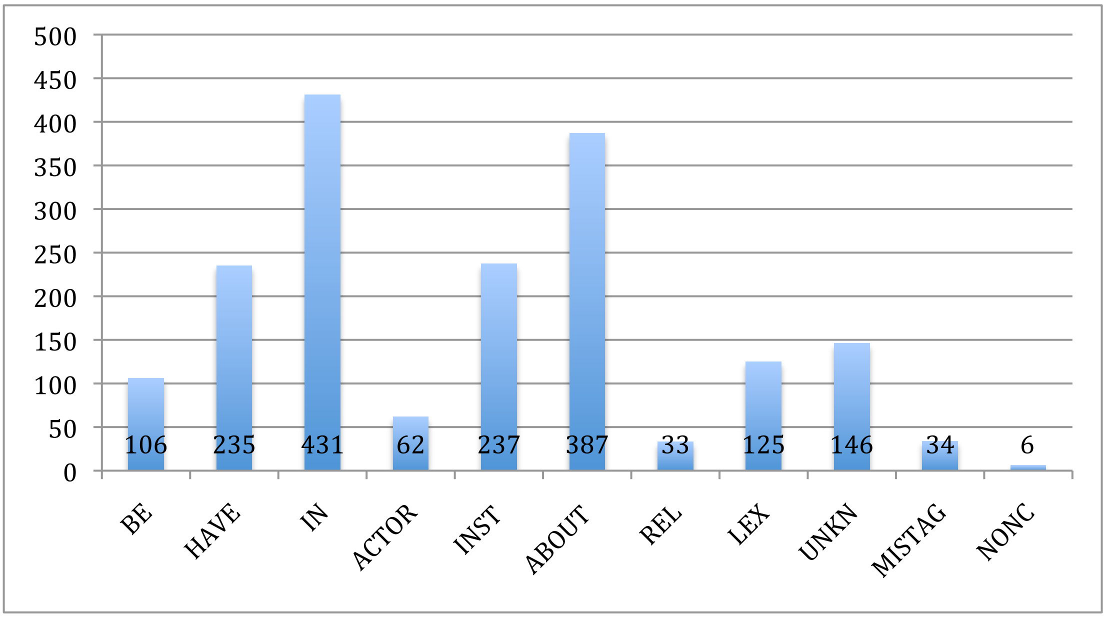


Figure 1. Class Distribution of Annotated Dutch Compounds.

### 4.2.3. 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. This calculation will be corrected for chance.

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 surely not equal for each class in this case. The Kappa measure will take the class distributions of the different annotators into account and thus provide a more reliable measure of annotation agreement (Boleda & Evert, 2009).

The inter-annotator agreement (IAA) on the categories of the 500 compounds was 60.2% (Kappa = 0.60). Although this is somewhat lower than other reported IAA's, e.g. Ó Séaghdha'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 categories together with the direction. The agreement here is 54%. The agreement on the complete annotated information (category, direction and rule) is 46.8%. It is essential to remark that Ó Séaghdha's guidelines "were not developed with the intention of maximising the distinctions between rules in the same category" (Ó Séaghdha, 2008:45).

Below, you will find the confusion matrix that compares the annotation of both annotators.

| | | **Annotator 1** | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | BE | HAVE | IN | ACTOR | INST | ABOUT | REL | LEX | UNKN | MISTAG | NONC |
| **Annotator 2** | BE | **20** | 3 | 2 | 0 | 3 | 2 | 1 | 3 | 1 | 1 | 1 |
| | HAVE | 2 | **40** | 16 | 1 | 5 | 9 | 1 | 5 | 6 | 0 | 0 |
| | IN | 2 | 9 | **87** | 2 | 1 | 11 | 0 | 1 | 3 | 0 | 0 |
| | ACTOR | 0 | 1 | 0 | **14** | 0 | 2 | 0 | 2 | 0 | 0 | 0 |
| | INST | 2 | 4 | 0 | 1 | **32** | 8 | 0 | 3 | 2 | 0 | 0 |
| | ABOUT | 4 | 7 | 9 | 6 | 9 | **60** | 0 | 3 | 2 | 0 | 0 |
| | REL | 1 | 1 | 1 | 0 | 1 | 2 | **2** | 3 | 2 | 1 | 0 |
| | LEX | 1 | 2 | 1 | 0 | 0 | 3 | 0 | **9** | 2 | 1 | 0 |
| | UNKN | 0 | 3 | 1 | 0 | 0 | 2 | 1 | 5 | **26** | 0 | 0 |
| | MISTAG | 0 | 1 | 0 | 0 | 0 | 3 | 0 | 1 | 2 | **11** | 1 |
| | NONC | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0** |

Table 1. Confusion Matrix of the Inter-Annotator Agreement

There are several factors that are likely to have contributed to our lower IAA. The most important being that the compounds were not accompanied by their context in our annotation process. This will cause a higher disagreement between the annotators because the context would normally constrain the possible interpretations of a compound.

Analysing this confusion matrix also shows us that there are certain categories that are rather often disagreed upon by the annotators. Remarkably, this interchangeable aspect works in both directions. The interchanged categories are colour coded and can be summed up as follows:

- IN & HAVE
- HAVE & ABOUT
- IN & ABOUT
- 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 probable that the first annotator was not ‘skilled’ enough in applying the guidelines. Because of the difficulty of this particular task, it may be necessary to put more time in the training of the annotators and do more test annotations together.

It was noted that 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%. (This area of the confusion matrix has been accentuated.) 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 68.16% with a kappa score of 67.90%. An overall agreement of 53.46% 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.

# 5. Experiment

Now that the compounds are annotated and ready, the next step is to start the actual semantic analysis experiment. The task of semantically analysing noun compounds has been reformulated as a classification task. Machine learning techniques will be used to train a classifier on our training instances. These instances are gathered in vectors that contain semantic information, based on context, of our compounds. The classifier will then assign categories to the compounds in the test set.

In a first section of this chapter, some theoretical assumptions about our experimentation methods will be discussed. The second part of this chapter explains the creation of the vectors that will be used to train our classifiers on. Section 5.3 then describes the machine learning techniques that are used to run our classification experiments. The results will be laid out in Chapter 6 and will be discussed in Chapter 7.

## 5.1. Theoretical Assumptions

In order to create the vectors for the compounds we must know what features we want to use. Section 3.2 of the Related Research chapter shows us that there are several possibilities. We have, however, already explained that our research is based on Ó Séaghdha's PhD thesis (2008). Ó Séaghdha uses two corpus-based proximity measures to assemble the compound vectors for the classifier. These are lexical similarity and relational similarity.

Our research will be restricted to the lexical similarity approach. The lexical similarity measure will provide us with the information that we need to construct the vectors with. This measure fits in a wider scheme of hypotheses about statistical methods and corpus research. Sections 5.1.1 to 5.1.3 will provide some background information on our research method.

### 5.1.1. Analogical Reasoning

The assumption underlying our entire research is that "the semantic relations in compounds […] can be identified through a process of analogical or similarity-based reasoning" (Ó Séaghdha, 2008:55). We thus assume that the semantic category of a compound (which represents the meaning between the compound constituents) can be predicted by comparison with compounds with similar meanings. "Or equivalently: the more similar two compounds are, the more likely they are to express similar semantic relations" (Ó Séaghdha, 2008:55). This assumption of analogy is also the basis of all statistical methods of classification, which is why we can interpret the compound interpretation task as a classification task (Ó Séaghdha, 2008).

This analogical hypothesis has already been wielded in numerous (computational) linguistic studies where it was proven a valuable research instrument (Ó Séaghdha, 2008).

### 5.1.2. Distributional Hypothesis

The analogical assumption can only be useful when we have a source of semantic information on the basis of which the compounds can be compared. The distributional hypothesis provides us with this semantic information (Ó Séaghdha, 2008).

The distributional hypothesis implies that "two words are semantically similar if they have similar patterns of co-occurrence with other words in some set of contexts" (Ó Séaghdha, 2008:59). The meaning of a word is thus implicitly present in the surrounding words (the context) of this word. Following this logic, we can assemble a vector of co-occurrence information of a word and consider it to represent an approximation of this word's meaning. Zellig Harris (1968) was one of the first to posit this idea; another theoretical work on this topic was by Firth (1957). Hinrich Schütze's (1992) research is well known for its early computational implementation of this idea. Other researchers computationally using the distributional hypothesis in its initial stages are Harper (1965) and Spärck-Jones (1964).

### 5.1.3.Lexical Similarity

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. For this first exploratory research into compound semantics in Dutch, we will adopt only the lexical similarity approach.

The lexical similarity 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.

## 5.2. Vector Creation

The theoretical notions of our experimentation methods have been defined in the previous section. What follows is a practical description of the vector construction.

### 5.2.1. General Outline

For every compound constituent in our annotated list, the context will have to be calculated. The Twente News Corpus (Ordelman, de Jong, van Hessen & Hondorp, 2007), a 340 million word Dutch corpus, was our source of co-occurrence contexts. A Python script was written to crawl this corpus and when coming across one of the constituents, the surrounding words were held in memory. 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 constructing constituent vectors. For every compound, we create a vector that contains the compound itself, its category, direction and rule (as annotated before), and then the relative frequencies for the 1000 most frequent words for the respective constituents. Compounds of which one or both of the constituents did not appear in the corpus, were excluded from our vector set.

Our final vector sets 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 1447 compounds in our dataset.

### 5.2.2. Different Variants of Training Data

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. This section will explain the design and purpose of the different variants of the vectors.

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 attested 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 vector creation is calculated by taking those context words that appear in the contexts of the 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. These approaches will respectively be abbreviated as ‘freq’ and ‘cont’ in the results chapter.

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, but you lose the morphological and syntax markers of the words that also might provide clues on the semantics of a word. The abbreviations to refer to these approaches are ‘lemma’ and ‘lex’.

The size of the co-occurrence context of the constituent is the third variation in our sets of training data. It will be interesting to see 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.

### 5.2.3. Principal Components Analysis

So far, we described a ‘bag of words’ approach where each token (lemma or lexical item) equals one attribute in the vector. Because the compound vector contains 2000 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 with the greatest variance 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 'PCA Module for Python' as implemented by Henning Risvik (2008). This module provides two algorithms to perform PCA on data: SVD and NIPALS. The SVD (singular value decomposition) algorithm is one of the basic algorithms to perform PCA. The NIPALS (non-linear iterative partial least squares) algorithm is used for very high-dimensional datasets because it allows the computation of only the first few PCs.

New datasets 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' vectors, we now also have SVD vectors for every variant. They have 100 attributes per compound (50 per constituent).

## 5.3. Machine Learning

The next step in our experimentation process is the actual machine learning experiments. A computational model is created and trained on the vectors (and their classes) that we constructed. This model will, by training on actual data and generalising this 'knowledge', learn the structure and regularities that are present in this data. It will connect certain aspects of the vectors to the classes these vectors belong to. It will eventually become able, up to a certain point, to predict the class of unseen instances.

There are several machine learning methods available. Our research will focus on support vector machines (SVMs) because they have been proven to work very well in many different natural language processing tasks, especially in text categorisation (e.g. Sebastiani, 2002). We will use instance-based learning for comparison. These two types of machine learning will be discussed below. After the discussion of the machine learning algorithms, the metrics and method that will be used to evaluate the experiments will be explained.

### 5.3.1. Support Vector Machines

The basic principle behind SVM algorithms is linearly separable binary classification. Let us assume a two-dimensional space where instances of two classes are situated according to their attributes in a vector. The SVM algorithm will try to linearly separate the instances of the two classes by finding a hyperplane that maximises the margin between these two classes. The instances that lie closest to this hyperplane are the support vectors. They support the location of the hyperplane. If the support vectors would be removed, the dividing hyperplane would change its position (Berwick, 2003; Fletcher, 2009).

When, as in many cases, the instances cannot be separated in their original space, the instances will be mapped onto a higher dimensional space by applying kernel functions. It is possible that the instances then become separable (Cortes & Vapnik, 1995).

When applying the SVM method to a multiclass problem, the algorithm will divide the problem in several binary problems and eventually combine the outcome of these split SVM tasks.

Our research will use the SMO implementation of the SVM algorithm that is provided in the WEKA Data Mining software (Witten, Frank & Hall, 2011). Automatic optimisation of the parameters will be performed by the CVParameterSelection function.

### 5.3.2. Instance-based Learning

Instance-based learning, or memory-based learning, is a type of machine learning where the algorithm does not create a model that represents the variance in the training data. Instead, the entire set of training data, all instances come across so far, is stored in memory. The performance of the classification algorithm is based on the comparison of new instances with the instances from the training data. The new instance will be assumed to belong to the same class as the training instance that is most similar (Daelemans & Van den Bosch, 2005).

The *k*-nearest neighbours algorithm (k-nn) is the most well known implementation of this type of machine learning. This algorithm does not use one most similar instance, but *k* similar instances (or neighbours) for comparison. The prevalent class of these *k* instances will be assigned to the test instance.

For our research, we will use the TiMBL IB1 algorithm, which is a *k*-nearest distance algorithm. This means that not the *k* nearest neighbours will be taken into account, but all the neighbours at the closest *k* distances to the considered instance. TiMBL is an open source memory-based machine learning software package that was developed by Daelemans & Van den Bosch (2005). Based on exploratory research, we used the IB1 algorithm with a *k*-value of 3 and no weighting of the attributes takes place.

### 5.3.3. Evaluation

This section will provide some details on the evaluation of our experiment. The main aspects of this topic are the method used for evaluation and the metrics that are used to present the model's accuracy.

#### 5.3.3.1. Evaluation Method: Cross-Validation

In order to maximise the potential of the training data of the experiment to generalise to the test data, it is desirable to train and test our classifier on as much data as possible. Cross-validation is a technique that serves exactly this purpose (Jurafsky & Martin, 2009).

In *k*-fold cross-validation, *k* different 'folds' of the same data are created. Every fold contains a different randomly chosen training and test set. The classifier is then trained and tested on each fold separately and the average results of these *k* runs are the measure of performance for this classifier on this data set (Jurafsky & Martin, 2009).

The most frequently used version of this technique is tenfold cross-validation in which 10 folds of the data set are created that each contain 90% train data and 10% test data. This method was also used in our compound classification experiments.

#### 5.3.3.2. Evaluation Metrics

Evaluation metrics are a means of representing the performance of a system. We will use accuracy, precision, recall and F-measure (or F-score) for this purpose. These are the standard evaluation metrics for research in computational linguistics. They are described in Jurafsky & Martin (2009), Compumine (2012) and Van Asch (2012).

Although our experiment holds a multiclass problem, the measures will first be presented as applied to a two-dimensional classification problem. A generalisation of these metrics, as well as the application to multi-class problems, will be presented at the end of this section.

Table 2 shows an example confusion matrix of a two-dimensional classification problem. Table 3 is the symbolic representation of the data for the evaluation of class A. TP stands for true positives; TN stands for true negatives, and FP and FN for false positives and negatives. In this representation 'positives' are those instances that are classified as the considered class (vs. negatives: instances classified as the other class). 'True' means that the positive/negative indication is true (vs. false: the positive/negative tag is incorrect);

| | | Predicted Class | |
|---|---|---|---|
| | | A | B |
| Known Class | A | 20 | 4 |
| | B | 3 | 15 |

Table 2. Example Confusion Matrix for a Two-Dimensional Classification Problem

| | | Predicted Class | |
|---|---|---|---|
| | | A | B |
| Known Class | A | TP | FN |
| | B | FP | TN |

Table 3. Symbolic Confusion Matrix for Evaluation of Class A

The *accuracy* of a classifier is the sum of all the correctly classified instances divided by the total sum of instances.

Accuracy = (TP + TN) / (TP + TN + FP + FN).

To be able to get more information on the performance of the classifier, other more specific measures can be computed. These measures are computed per class and can be averaged to tell something about the entire system's performance.

The *precision* for a class is actually the accuracy measure calculated for this class only. This is calculated by taking the number of correctly classified instances for this class and dividing it by the number of instances that are classified as the considered class.

Precision for A = TP / (TP + FP)

The *recall* for a class is a measure that represents the ability of a model to select instances of a class from a data set. This measure can be computed by dividing the number of correctly classified instances of the considered class by all the instances of this class.

Recall for A = TP / (TP + FN)

The F-score is a combination of the former two measures into a single metric. The importance of precision and recall can be weighted in this combination, depending on the goal of your system. In most cases however, precision and recall are equally weighted. This yields the following formula for F-score calculation.

F-score = ( 2 x Precision x Recall ) / ( Precision + Recall )

The above three metrics are applicable to one class in a problem. To obtain a global result for the experiment, a weighted average of the individual precisions, recalls and F-scores will be calculated. There are two ways of averaging these evaluation measures: *macro-average* and *micro-average* (Van Asch, 2012). These can be distinguished as follows:

> Macroaveraging gives equal weight to each class, whereas microaveraging gives equal weight to each per-document classification decision. […] [L]arge classes dominate small classes in microaveraging. […] Microaveraged results are therefore really a measure of effectiveness on the large classes in a test collection. To get a sense of effectiveness on small classes, you should compute macroaveraged results (Manning et al., 2008:280-281).

When dealing with a multiclass problem, as in the compound semantics analysis, a ‘positive’ instance is then an instance that is classified as belonging to the considered class; a ‘negative’ instance can belong to any other class in the experiment.

For certain results, the statistical significance will be calculated. The calculation of a statistical significance shows whether two results are more different than what could have been caused by chance. A significance value *p* that is below 0.05 is considered significant.

The Approximate Randomization Testing script (art.py[4]) was used in our research to compute statistical significance. This script was written by Vincent Van Asch from the CLiPS Computational Linguistics research group at the University of Antwerp.

The results of our experiments using the support vector machines and instance-based learning on the Dutch compound data will be presented in Chapter 6.

[4] This script is publicly available at: http://www.clips.ua.ac.be/scripts/art

# 6. RESULTS

Having discussed some background topics about compounding and explained the entire experimental setup, you will now be provided with the results.

This chapter 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 in the next section. An error analysis can be found in the fourth section of this chapter. Finally, some additional experiments are discussed in section 6.4.

## 6.1. Main results

To obtain the following results on our classification task for the semantics of Dutch compounds, the WEKA (Witten, Frank & Hall, 2011) and TiMBL (Daelemans & Van den Bosch, 2005) software packages were provided with the twelve variants of our vectors (as described in section 5.2.2). The SMO algorithm (WEKA's SVM implementation) 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 vectors.

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 (1447). This number represents the accuracy that can be obtained by always guessing IN as the output class.

Table 4 presents the micro-average results achieved with the SMO classifier.

| **Variants** | | | **BOW** | | | **PCA - SVD** | | |
|---|---|---|---|---|---|---|---|---|
| | | | Precision | Recall | F-Score | Precision | Recall | F-Score |
| Freq | Lemma | 3 | 47.7 | 47.5 | 47.6 | 44.5 | 48.1 | 44.6 |
| Freq | Lex | 3 | 47.6 | 48.0 | 47.8 | 41.7 | 46.2 | 41.7 |
| Cont | Lemma | 3 | **49.5** | **48.8** | **49.0** | 45.2 | 47.8 | 45.1 |
| Cont | Lex | 3 | 46.7 | 47.0 | 46.8 | 43.7 | 46.8 | 42.9 |
| Freq | Lemma | 5 | 46.6 | 46.6 | 46.5 | 45.4 | 48.2 | 44.2 |
| Freq | Lex | 5 | 47.7 | 48.0 | 47.8 | 43 | 47.6 | 43.6 |
| Cont | Lemma | 5 | 45.7 | 45.5 | 45.5 | **45.8** | **48.4** | **45.2** |
| Cont | Lex | 5 | 47.8 | 48.4 | 48.0 | 44.4 | 48.4 | 43.9 |
| Freq | Lemma | 10 | 47.0 | 47.0 | 46.9 | 45.5 | 47.9 | 42.9 |
| Freq | Lex | 10 | 47.2 | 47.7 | 47.4 | 44.2 | 48.4 | 42.5 |
| Cont | Lemma | 10 | 46.4 | 46.3 | 46.3 | 44.2 | 47.9 | 42.8 |
| Cont | Lex | 10 | 47.4 | 48.0 | 47.6 | 42.3 | 47.8 | 41.8 |

Table 4. Micro-Average SMO Results on BOW and PCA Vectors using 10-fold Cross-Validation.

The results in Table 4 clearly show a significant improvement over the most frequent class baseline. The BOW approach reaches better results, with a maximum of 49% F-score, than the PCA 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 still seems to be outperformed by the BOW approach. When taking a closer look at the results of the best performing PCA and BOW experiments, this statement appears to be true. 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 on 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.

The PCA vectors were also used for experiments with TiMBL's IB1 algorithm. The results of these experiments can be found in Table 5.

| **Variants** | | | **PCA – SVD** | |
|---|---|---|---|---|
| | | | Accuracy | F-Score |
| Freq | Lemma | 3 | 46.7 | 44.8 |
| Freq | Lex | 3 | 48.3 | 46.0 |
| Cont | Lemma | 3 | 45.8 | 44.0 |
| Cont | Lex | 3 | 49.1 | 47.2 |
| Freq | Lemma | 5 | **49.7** | **47.7** |
| Freq | Lex | 5 | 49.6 | 47.2 |
| Cont | Lemma | 5 | 47.5 | 45.8 |
| Cont | Lex | 5 | **49.7** | **47.7** |
| Freq | Lemma | 10 | 48.3 | 46.4 |
| Freq | Lex | 10 | 47.0 | 44.9 |
| Cont | Lemma | 10 | 48.6 | 46.6 |
| Cont | Lex | 10 | 49.6 | 47.4 |

Table 5. IB1 Results on the PCA Vectors.

The F-scores achieved by the IB1 algorithm on the PCA vectors range from 44% to 47.7%. The average F-score of all the variants is 46.3%. This is a remarkably higher result than the SMO algorithm.

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

After performing these experiments, we noticed a methodological problem in our ‘Cont’ method. Due to the calculation of the 10,000 most frequent words on the entire set of constituents, our instances are not independent which is a necessary condition to test generalization. This may lead to overfitting. The ‘Freq’ method calculates its most frequent words list on the entire corpus. This method is data independent and thus does not cause any overfitting. In order to assess the effect on our results of this possible overfitting, we performed another experiment in which we use a fixed training and test set. The most frequent words list for the ‘Cont’ method is then calculated on the training set only and applied on the test set. For a better comparison, we also performed the train/test experiment using the ‘Freq’ method. The results of this experiment using the BOW approach can be found in Table 6.

| **Variants** | | **CONT** | | | **FREQ** | | |
|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F-Score | Precision | Recall | F-Score |
| Lemma | 3 | 50.5 | 50.4 | 49.3 | 52.9 | 50.4 | 50.1 |
| | 5 | 48.9 | 48.9 | 48.2 | 49.8 | 50.4 | 49.5 |
| | 10 | 52.7 | 52.6 | 51.6 | 57.6 | 56.2 | 55.5 |
| Lex | 3 | 56.4 | 54.7 | 54.1 | 58.3 | 56.2 | 55.8 |
| | 5 | 52.8 | 51.8 | 51.2 | **57.1** | **56.9** | **56.8** |
| | 10 | **56.3** | **56.2** | **55.4** | 53 | 52.6 | 52.5 |

Table 6. Micro-Average SMO Results on BOW Vectors using Fixed Training and Test Set.

We will further look into all our results in the next section.

### 6.2. Tendencies

Tables 7 and 8 are included in this section to illustrate some of the tendencies that can be noticed in the main experimental results.

| | | Avg. F-Score BOW | Avg. F-Score PCA |
|---|---|---|---|
| Number of Context Words (Left and Right) | 3 | **47.8** | 43.5 |
| | 5 | 47.0 | **44.2** |
| | 10 | 47.1 | 42.5 |
| Manner of Context Calculation | Freq | 47.5 | 43.2 |
| | Cont | 47.2 | 43.6 |
| Type of Corpus Elements | Lemma | 46.9 | **44.1** |
| | Lex | **47.5** | 42.7 |

Table 7. Average SMO F-scores for Different Experimental Aspects.

| | | Avg. F-Score PCA |
|---|---|---|
| Number of Context Words (Left and Right) | 3 | 45.5 |
| | 5 | **47.1** |
| | 10 | 46.3 |
| Manner of Context Calculation | Freq | 46.5 |
| | Cont | 46.4 |
| Type of Corpus Elements | Lemma | 45.8 |
| | Lex | **47.1** |

Table 8. Average IB1 F-scores for Different Experimental Aspects.

A first observation of these tables teaches us that there is hardly any difference in results due to the manner of context calculation. Our secondary experiment (see Table 6) with improved methodology on a fixed training and test set gives average F-scores of 51.63% for the 'Cont' method and 53.36% using the 'Freq' method. Although evaluation with training and test sets is less accurate than 10-fold cross-validation, the hypothesis that the 'more context' method would perhaps raise the results has hereby been proven faulty.

Since the results of our secondary experiment confirm the results of our main experiment that the 'Cont' method does not live up to our expectations, we will further only discuss the results of our main experiment; the reason being that we consider 10-fold cross-validation a more reliable evaluation method than training and test sets.

The number of context words that are being taken into account does 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 and the IB1 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 and the IB1 PCA results both 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.

### 6.3. Result Analysis

In this section, a result analysis of the classification of the best performing experiment will be 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 yielded the best results: BOW Cont Lemma 3.

| | | Classifier | | | | | |
|---|---|---|---|---|---|---|---|
| | | INST | HAVE | ABOUT | IN | ACTOR | BE |
| Annotation | INST | **106** | 35 | 39 | 37 | 3 | 15 |
| | HAVE | 24 | **90** | 46 | 49 | 6 | 18 |
| | ABOUT | 65 | 53 | **210** | 36 | 10 | 10 |
| | IN | 41 | 58 | 43 | **248** | 15 | 23 |
| | ACTOR | 9 | 5 | 10 | 6 | **28** | 4 |
| | BE | 14 | 25 | 18 | 23 | 1 | **24** |

Table 9. Confusion Matrix of the Classification with the Best Results: SMO BOW Cont Lemma 3.

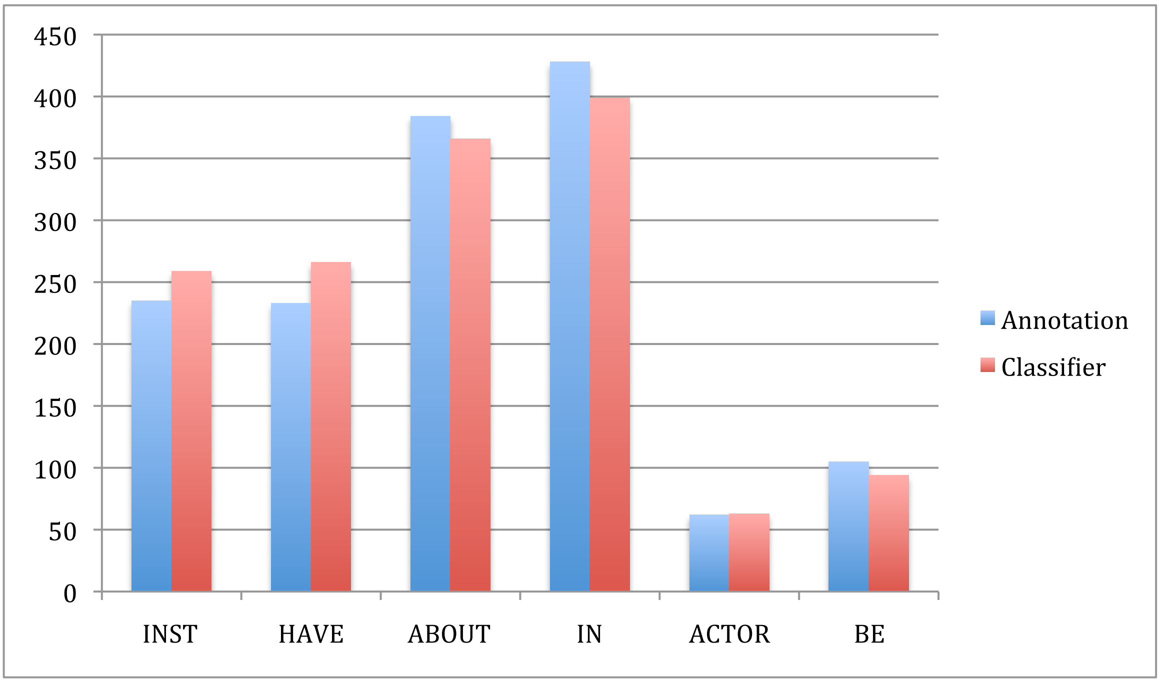


Figure 2. Comparison of Class Distributions between Annotation and Best Classifier.

| | Precision | Recall | F-Score |
|---|---|---|---|
| INST | 40.9 | 45.1 | 42.9 |
| HAVE | 33.8 | 38.6 | 36.1 |
| ABOUT | 57.4 | 54.7 | 56 |
| IN | 62.2 | 57.9 | 60 |
| ACTOR | 44.4 | 45.2 | 44.8 |
| BE | 25.5 | 22.9 | 24.1 |
| Weighted Avg. | 49.5 | 48.8 | 49 |

Table 10. Accuracies by Class of the Confusion Matrix in Table 6.

Table 9 presents the confusion matrix of the best classifier; this is the SMO algorithm on the BOW Cont Lemma 3 data set. There are many misclassifications, which is normal with an F-score of 49%, but there seem to be no strong tendencies 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 10 provides us with the results by class that this classifier achieves. It is noticeable that classes with higher frequencies reach a higher accuracy, which makes sense since there is more training information on this class available because of its higher frequency. The BE class has a rather low frequency and has the lowest accuracy with an F-score of 24.1%. However, the ACTOR category, which has the lowest frequency, does have the third highest accuracy (44.8%). This indicates that the ACTOR category is easier to learn by the classifier than the classes with higher frequencies.

### 6.4. 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 dubiously 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* 'spa treatment' (lit. bath 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.

### 6.5. Additional Experiments

Some minor side experiments were also performed on the data. These can be considered as precursors of further research on these matters.

### 6.5.1. NIPALS Algorithm

Principal Components Analysis, as implemented in the PCA Module for Python (Risvik, 2008), can be calculated by two algorithms. For the purposes of our research, we used the SVD algorithm, which is the basic PCA algorithm. The NIPALS algorithm is developed for large multidimensional data sets. It may thus be interesting to try and use this algorithm on our best SVD result and compare these two algorithms.

Table 11 provides the results of the SVD and NIPALS algorithm on the Cont Lemma 5 dataset.

| | Precision | Recall | F-score |
|---|---|---|---|
| SVD | 45.8 | 48.4 | 45.2 |
| NIPALS | 46.3 | 48.8 | 45.9 |

Table 11. Comparison of Results with SVD or NIPALS Algorithm.

It appears that the NIPALS algorithm can present somewhat higher results than the SVD algorithm. These differences are, nevertheless, not statistically significant. The significance value p for the micro-average F-score is 0.419, which is higher than 0.05 and thus not statistically significant. It may still be worth looking more into the differences between these algorithms to make a choice between them for further research.

### 6.5.2. Eight 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.

|  | 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 |
| Micro-average | 45.2 | 41 |

Table 12. Comparison of F-Scores per Class with 6 or 8 Categories.

Table 12 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.

### 6.5.3. Rule Induction Classifier

A final additional experiment was to investigate the performance of a rule induction classifier on the best performing BOW data: Cont Lemma 3. More importantly, it will be interesting to see if the algorithm creates some interpretable rules based on the unigrams in the bag of words. WEKA's JRip algorithm (Witten, Frank & Hall, 2011) was used with its default parameters for this purpose.

An F-score of 36.5% was reported. This is not a large improvement from the 29.5% baseline, but let us have a look at the rules anyway.

JRip constructs 16 rules that are to be followed in order. The most noticeable aspect of these rules is that they completely ignore the BE category. The first four rules will classify instances as ACTOR. Then there are three rules for HAVE and three rules for INST. The ABOUT category has five rules governing its instances. The rest of the instances is automatically categorised as IN. There is no rule that allows for instances to be classified as BE. Although this may be a small class, this is already detrimental for the system's usefulness. However, this problem may be partially solved by optimising the algorithm's parameters. It does seem appropriate for the default category to be IN, since this is the largest category in the training data.

When looking at the unigrams used in the rules, some minor tendencies may be identified. It appears that the ACTOR category is often associated with verbs, e.g. *zeggen* (to say), *beginnen* (to begin)*,* *betalen* (to pay) and *spreken* (to speak/talk). The HAVE category has pronouns such as *we, elkaar* (each other) and *beide* (both) that may point towards possession. The INST category has nouns like *oog* (eye) and *hand* that can be seen as instruments of actions in some way. The verb *zien* (to see, which fits with the eye) and the preposition *als* (as) may also be indicators of instrumental actions.

On the other hand, there is quite some evidence for overfitting in these rules. Proper names (*Peter, Utrecht)* and numerals (*zestig* ‘sixty’*)* cannot really add any semantic value to these rules.

# 7. DISCUSSION

## 7.1. Summary

This research focused on the semantic analysis of Dutch noun compounds by using computational classification methods. The noun compounds were semantically annotated in advance. These annotations were performed using the provided guidelines that describe different 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.

The introduction in Chapter 1 describes the context and the research goals of our experiment and the applications that could possibly benefit from an automatic semantic analyser of compounds. The research goals are identified as follows:

- The creation of a semantically annotated list of compounds.
- Exploratory research on the feature creation for the classification of Dutch compounds and presenting initial performance results.
- The investigation of the performance of the experiments using measures of dimensionality reduction, like Principal Components Analysis (PCA).

Chapter 2 provides a background overview of the theoretical linguistic research on compounds. A first section focused on the problematic definition of compounding. The second part of this chapter deals with the semantic aspects of compounding.

The related research is presented in Chapter 3. We noted that most annotation schemes used in similar research are based on a paraphrasing predicate of the relation between the compound constituents. Different methods of feature selection were then discussed. The most prominent of which are the corpus-based methods that use proximity features and the taxonomy-based methods that often use WordNet.

In order to provide our compounds with a semantic annotation, we needed an annotation scheme fit for this task. We adopted the scheme and guidelines posited by Ó Séaghdha in 2008. This scheme was slightly adapted to be appropriate for the Dutch compounds. In the annotation process, an inter-annotator agreement of 60.2% was achieved. Chapter 4 deals with all aspects of the annotation guidelines and process.

Chapter 5 contains the description of our experimental setup. The principles of analogical reasoning and the distributional hypothesis are first laid out, after which the measure of lexical similarity is explained. This measure uses the semantics of the separate constituents of compounds to make a decision on the overall semantics of the compounds.

The next section describes the creation of our vectors. These vectors contain information on the context in which the constituents of the compound occur. Different variants of these vectors are created to be able to compare between different experimental aspects. Also, Principal Components Analysis was performed on variants of the vectors. This method reduces the number of vectors while maintaining the variance in this data.

The last section of Chapter 5 provides background information about the machine learning algorithms used in our classification experiments and about the cross-validation method and evaluation metrics used.

In Chapter 6, our experimental results are presented. We achieved a best result of 49% F-score on the BOW data which significantly outperforms the 29.5% most frequent class baseline. The PCA approach also reaches significant results with a best F-score of 45.2%. The BOW still outperforms the PCA approach on the recall of the smaller classes. We also present analyses of the results by looking for tendencies and by looking at the results in more detail.

## 7.2. Conclusions

The research goals that were posited in the introduction will now be revisited and discussed. A state of affairs on the research goals will be presented and some conclusions can be drawn.

- The creation of a semantically annotated list of compounds.

A semantically annotated list of compounds was indeed created. The annotated list contains 1802 Dutch noun compounds that are each annotated with one of eleven categories. Of this list, 1447 compounds belong to one of the six semantically specific classes that can be used for machine learning.

The inter-annotator agreement was 60.2%, which is a good initial result. This score does not reach the same heights as the best agreements reported for English systems, but this may be because our compounds were not annotated in context.

- Exploratory research on the feature creation for the classification of Dutch compounds and presenting initial performance results.

This initial research on the automatic semantic analysis of Dutch noun compounds reached significant results. Our best performing experiment had a micro-average F-score of 49%, 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%) reached on English compounds using the same method (Ó Séaghdha, 2008). The BOW approach appears to consistently outperform the PCA approach, but the IB1 experiment also yields better results on the PCA data than the SMO algorithm.

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 would no longer only describe the constituent, but also include 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 the difference can be disputed. The SMO BOW experiments and the IB1 PCA experiments showed better performance using the lexical items, the SMO PCA experiments preferred using the lemmas. However, these are tendencies that can be seen in the experiment averages. These are not visible in individual experiments.

The error analysis gives an indication that the annotator will need more training, but it also hints at a rather good performance of our classifier given the number of compounds that were misannotated but correctly classified.

- The investigation of the performance of the experiments using measures of dimensionality reduction, like Principal Components Analysis (PCA).

The results of our experiments using the PCA approach turned out not only to be significant 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 significantly lower than that of the BOW approach.

These results are promising because they might allow us to create smaller data sets in our vectors, 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.

### 7.3. Further Research

Future research will have to 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.

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 a must in future research. These measures should raise the agreement, and hopefully 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 very 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.

The PCA approach might need some further exploration, but it appears to be working fine on this data. The NIPALS algorithm can be more looked into, and possibly some other implementations too.

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.

Analogous to previous research on English compound semantics, combining our current vectors with (hyponym) information from a WordNet-like semantic network (for Dutch: CORNETTO) should also have a positive impact on our results.

# BIBLIOGRAPHY

Barker, K., & S. Szpakowicz. (1998). 'Semi-Automatic Recognition of Noun-Modifier Relationships'. In: *Proceedings of the 17th International Conference on Computational Linguistics.* San Francisco, CA: Morgan Kaufmann, 96-102.

Bauer, L. (2009). 'Typology of compounds'. In: Lieber, R., & P. Štekauer (eds.). *The Oxford Handbook of Compounding*. Oxford: Oxford University Press, 343-356.

Benczes, R. (2006). *Creative Compounding in English: The semantics of metaphorical and metonymical noun-noun combinations.* Philadelphia: John Benjamins Publishing Company.

Berwick, R. (2003). *An Idiot's Guide to Support Vector Machines (SVMs).* Cambridge, MA: MIT. <http://www.cs.ucf.edu/courses/cap6412/fall2009/papers/Berwick2003.pdf> (28/07/2012).

Boleda, G., & S. Evert. (2009). 'Computational Lexical Semantics: Inter-Annotator Agreement'. *European Summer School in Logic, Language and Information (ESSLLI-09)*. Bordeaux: Association for Logic, Language and Information (FoLLI). <http://clseslli09.files.wordpress.com/2009/07/02_iaa-slides1.pdf> (03/08/2012).

Booij, G. (1996). 'Verbindingsklanken in samenstellingen en de nieuwe spellingregeling'. *Nederlandse Taalkunde* 2, 126-134.

Booij, G. (2009). 'Compounding and construction morphology'. In: Lieber, R., & P. Štekauer (eds.). *The Oxford Handbook of Compounding*. Oxford: Oxford University Press, 201-216.

Booij, G. (2010). *Construction Morphology.* Oxford: Oxford University Press.

Butnariu, C., Kim, S.N., Nakov, P., Ó Séaghdha, D., Szpakowicz, S., & T. Veale. (2010). 'SemEval-2010 Task 9: The Interpretation of Noun Compounds Using Paraphrasing Verbs and Prepositions.' In: *Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010.* Uppsala: Association for Computational Linguistics, 39-44.

Compumine. (2012). *Evaluating a Classification Model – What does precision and recall tell me?* Sweden: Compumine. <http://www.compumine.se/web/public/newsletter/20071/precision-recall> (02/08/2012).

Coolen, R. (1994). *The Semantic Processing of Isolated Novel Nominal Compounds*. Nijmegen: Quickprint. Proefschrift Katholieke Universiteit Nijmegen.

Cortes, C., & V. Vapnik. (1995). 'Support-Vector Networks'. *Machine Learning* 20 (3), 273-297.

Daelemans, W., & A. Van den Bosch. (2005). *Memory-Based Language Processing*. Cambridge, UK: Cambridge University Press.

Don, J. (2009). 'IE, Germanic: Dutch'. In: Lieber, R., & P. Štekauer (eds.). *The Oxford Handbook of Compounding*. Oxford: Oxford University Press, 370-385.

Evert, S. (2010). 'Distributional Semantic Models'. In: *Proceedings of Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, CA: Association for Computational Linguistics.

Finin, T.W. (1980). 'The Semantic Interpretation of Compound Nominals'. In: *Proceedings of the First Annual National Conference on Artificial Intelligence.* AAAI Press.

Fletcher, T. (2009). *Support Vector Machines Explained.* London: UCL. <http://www.tristanfletcher.co.uk/SVM%20Explained.pdf> (28/07/2012).

Giegerich, H. (2009). 'The English Compound Stress Myth.' *Word Structure* 2, 1-17.

Girju, R., Moldovan, D., Tatu, M., & D. Antohe. (2005). 'On the Semantics of Noun Compounds.' *Computer Speech and Language* 19, 479-496.

Girju, R., Nakov, P., Nastase, V., Szpakowicz, S., Turney, P., & D. Yuret. (2007). 'SemEval-2007 Task 04: Classification of Semantic Relations between Nominals'. In: *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*. Prague: Association for Computational Linguistics, 13-18.

Guevara, E., & S. Scalise. (2009). 'Searching for universals in compounding'. In: Scalise, S., Magni, E., & A. Bisetto (eds.). *Universals of language today.* Dordrecht: Springer, 101–128.

Harper, K.E. (1965). 'Measurement of Similarity Between Nouns'. In: *Proceedings of the 1965 International Conference on Computational Linguistics (COLING-65).* New York, NY: International Committee on Computational Linguistics, 1-23.

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

Jurafsky, D., & J.H. Martin. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (Second Edition).* Upper Saddle River, NJ: Pearson.

Katamba, F. (1994). *English Words.* London: Routledge.

Kim, S.N., & T. Baldwin. (2005). 'Automatic Interpretation of Noun Compounds Using WordNet Similarity'. In: *Second International Joint Conference on Natural Language Processing.* Jeju, Korea: Asian Federation of Natural Language Processing, 945-956.

Kim, S.N., & T. Baldwin. (2006). ‘Interpreting Semantic Relations in Noun Compounds via Verb Semantics’. In: *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions.* Sydney: Association for Computational Linguistics, 491-498.

Krott, A., Schreuder, R., Baayen, R. H., & W.U. Dressler. (2007). ‘Analogical effects on linking elements in German compounds’. *Language and Cognitive Processes* 22, 25-57.

Lapata, M., & F. Keller. (2004). ‘The Web as a Baseline: Evaluating the Performance of Unsupervised Web-based Models for a Range of NLP Tasks’. In: *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics.* Boston: Association for Computational Linguistics, 121-128.

Lauer, M. (1995). *Designing Statistical Language Learners: Experiments on Noun Compounds.* PhD Thesis, Macquarie University, Australia.

Lees, R.B. (1970). ‘Problems in the Grammatical Analysis of English Nominal Compounds. In: Bierwisch, M. & K.E. Heidolph (eds.). *Progress in linguistics.* The Hague: Mouton de Gruyter.

Lieber, R., & P. Štekauer. (2009). ‘Introduction: status and definition of compounding’. In: Lieber, R., & P. Štekauer (eds.). *The Oxford Handbook of Compounding*. Oxford: Oxford University Press, 3-18.

Lüdeling, A. (2009). *Neoclassical word-formation*. Berlin: Universität zu Berlin.

Manning, C., Raghavan, P., & H. Schütze. (2008). *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press.

Miller, G.A. (1995). ‘WordNet: A Lexical Database for English’. *Communications of the ACM* 38 (11), 39-41.

Moldovan, D., Badulescu, A., Tatu, M., Antohe, D., & R. Girju. (2004). ‘Models for the Semantic Classification of Noun Compounds’. In: *Proceedings of the HLT-NAACL Workshop on Computational Lexical Semantics.* Boston, MA: Association for Computational Linguistics, 60-67.

Nakov, P. (2008). ‘Noun Compound Interpretation Using Paraphrasing Verbs: Feasibility Study’. In: *Proceedings of the 13th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA’08).*

Nastase, V., Sayyad-Shirabad, J., Sokolova, M., & S. Szpakowicz. (2006). ‘Learning Noun-Modifier Semantic Relations with Corpus-based and WordNet-based Features’. In: *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06).* Boston, MA: American Association for Artificial Intelligence, 781-787.

Nederlandse Taalunie. (2011). *Bronbestand Woordenlijst Nederlandse Taal 2005*. <http://www.inl.nl/tst-centrale/nl/producten/lexica/bronbestand-woordenlijst-nederlandse-taal-2005/7-26> (20/07/2012).

Ogden, W.C., & P. Bernick. (1997). 'Using Natural Language Interfaces'. In: Helander, M., T.K. Landauer, & P. Prabhu (eds.). *Handbook of Human-Computer Interaction: Second, Completely Revised Edition*. The Netherlands: Elsevier Science, 137-162.

Ordelman, R., de Jong, F., van Hessen, A., & H. Hondorp. (2007). 'TwNC: a Multifaceted Dutch News Corpus'. *ELRA Newsletter* 12, 3-4.

Ó Séaghdha, D. (2007). 'Annotating and Learning Compound Noun Semantics'. In: *Proceedings of the ACL 2007 Student Research Workshop*. Prague: Association for Computational Linguistics, 73-78.

Ó Séaghdha, D. (2008). *Learning Compound Noun Semantics*. PhD Thesis, University of Cambridge.

Ó Séaghdha, D. (2009). 'Semantic classification with WordNet Kernels'. In: *Proceedings of NAACL HLT 2009: Short Papers*. Boulder, Colorado: Association for Computational Linguistics, 237-240.

Ó Séaghdha, D., & A. Copestake. (2007). 'Co-occurrence Contexts for Noun Compound Interpretation'. In: *Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*. Prague: Association for Computational Linguistics, 57-64.

Ó Séaghdha, D., & A. Copestake (2008). 'Semantic Classification with Distributional Kernels'. In: *Proceedings of the 22nd International Conference on Computational Linguistics (COLING-08)*. Manchester: Association for Computational Linguistics, 649-656.

Plag, I. (2003). *Word-Formation in English*. Cambridge: Cambridge University Press.

Plag, I. (2006). 'The variability of compound stress in English: structural, semantic and analogical factors.' *English Language and Linguistics* 10.1. Cambridge: Cambridge University Press, 143-162.

Risvik, H. (2008). *PCA Module for Python*. University of Oslo. <http://folk.uio.no/henninri/pca_module/> (27/05/2012).

Rosario, B., & M. Hearst. (2001). 'Classifying the Semantic Relations in Noun Compounds via a Domain-Specific Lexical Hierarchy'. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2001)*. Pittsburgh, PA: Association for Computational Linguistics, 82-90.

Salaberry, R. (1996). 'A Theoretical Foundation For Development of Pedagogical Tasks in Computer Mediated Communication'. *CALICO Journal* 14 (1): 5-34.

Scalise, S., & A. Bisetto. (2009). 'The classification of compounds'. In: Lieber, R., & P. Štekauer (eds.). *The Oxford Handbook of Compounding*. Oxford: Oxford University Press, 34-53.

Schütze, H. (1992). 'Dimensions of Meaning'. In: *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*. Los Alamitos, CA: IEEE Computer Society Press, 787-796.

Sebastiani, F. (2002). 'Machine Learning in Automated Text Categorization'. *ACM Computing Surveys* 34, 1-47.

SIL International. (2003). 'lexeme.' In: *LinguaLinks Library*. <http://www.sil.org/linguistics/GlossaryOfLinguisticTerms/WhatIsALexeme.htm> (04/04/2012).

Smith, L.I. (2002). *A Tutorial on Principal Components Analysis.* New Zealand: University of Otago. <http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf> (28/07/2012).

Spärck-Jones, K. (1964). *Synonymy and Semantic Classification*. PhD Thesis, University of Cambridge, UK.

Spencer, A. (2005). 'Word-Formation and Syntax'. In: Štekauer , P., & R. Lieber (eds.). *Handbook of Word-Formation*. Dordrecht, The Netherlands: Springer, 73-97.

Tratz, S., & E. Hovy. (2010). 'A Taxonomy, Dataset, and Classifier for Automatic Noun Compound Interpretation'. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.* Uppsala: Association for Computational Linguistics, 678-687.

Turney, P.D. (2006). 'Similarity of Semantic Relations'. *Computational Linguistics* 32 (3), 379-416.

Van Asch, V. (2012). *Macro- and Micro-Averaged Evaluation Measures.* Belgium: University of Antwerp. <http://www.clips.ua.ac.be/~vincent/pdf/microaverage.pdf> (05/08/2012).

Wijaya, D.T., & P. Gianfortoni. (2011). ' "Nut Case: What does it mean?": Understanding Semantic Relationship between Nouns in Noun Compounds through Paraphrasing and Ranking the Paraphrases'. In: *Proceedings of the 1st International Workshop on Search and Mining Entity-Relationship Data (SMER-11).* Glasgow, UK.

Winograd, T., & F. Flores. (1986). *Understanding Computers and Cognition: A New Foundation for Design*. Norwood, NJ: Ablex Publishing Corporation.

Witten, I.H., Frank, E., & M.A. Hall. (2011). *Data Mining: Practical Machine Learning Tools and Techniques (Third Edition).* Burlington, MA: Morgan Kaufmann.

Wolfram, S. (2010). 'Programming with Natural Language is Actually Going to Work'. *Wolfram Blog*. <http://blog.wolfram.com/2010/11/16/programming-with-natural-language-is-actually-going-to-work/> (02/08/2012).

# APPENDIX: ANNOTATION GUIDELINES FOR COMPOUND NOUNS IN ENGLISH AND DUTCH

These guidelines were taken and adapted from Ó Séaghdha's PhD thesis 'On Compound Semantics' (2008). They are developed to be able to describe the semantic relation between the constituents of two-noun compounds. We have only annotated those compounds that are not in the dictionary, but of which the constituent nouns can in fact be found in the dictionary. If a compound already has a gloss, we do not have to analyse it to find its meaning, but we do need to know the meaning of each constituent to be able to find the compound meaning. This means that a lot of common, lexicalised and exocentric compounds are excluded from the annotation. These compounds will be removed from the annotation data by crosschecking the data with a dictionary before the annotation commences. Should we still encounter such compounds in our data, rule 1.4 explains what to do with them.
More details on the adaptation of these guidelines can be found in chapter 4 of my master thesis.

## 1. General Guidelines

The task is to annotate each compound noun N1 N2 with regard to the semantic relation that holds between the constituent nouns N1 and N2. It is assumed that compounds are either copulative or semantically right-headed.

**Rule 1.1** *The general annotation format is* <RELATION,DIRECTION,RULE>.

RELATION is one of the 10 relation labels defined in section 2 of these guidelines. DIRECTION specifies the order of the constituent nouns in the chosen relation's argument structure – in particular, direction will have the value 1 if the first noun in the compound (N1) fits in the first noun slot mentioned in the rule licensing the chosen relation, and will have value 2 if the second noun in the compound (N2) fits in the rule's first noun slot. RULE is the number of the rule licensing the relation. For example:

*water fern*
IN,2,2.1.3.1
This aquatic water fern is a rosette plant which has dense, fibrous roots

*enemy provocation*
ACTOR,1,2.1.4.1
The army said at the weekend that troops had reacted to enemy provocations and intervened to protect local citizens

In the case of *water fern* the IN relation is licensed by Rule 2.1.3.1 *N1/N2 is an object spatially located in or near N2/N1*. Mapping the compound's constituent nouns onto the rule definition, we see that the first slot (N1/N2 is. . . ) is filled by N2 *fern* and hence the direction is 2. For the categories BE, REL, LEX, UNKNOWN, MISTAG and NONCOMPOUND there is no salient sense of directionality, so it need not be annotated:

> *cedar tree*
> BE,2.1.1.1
> On rising ground at the western end of the churchyard of St Mary's at Morpeth in Northumberland stands, sheltered by cedar trees, a funerary monument

In practice, we will assign every compound a direction to have uniformity in the encoding. Every compound from a category that has no sense of directionality (see above) will be encoded with direction 1.
In the examples of section 2 you will find the direction of the example in brackets behind the compound.

**Rule 1.2** *Each compound is presented with its sentential context and should be interpreted within that context. Knowledge of other instances of the compound type are irrelevant.*

A given compound type can have different meanings in different contexts. A *school book* is frequently a book read IN school, but it could also be a book ABOUT school. A *wood table* might be a table that IS wood (BE), but it might also be a table for chopping wood on (IN). The intended meaning of a compound is often clarified by the sentence it appears in.

**Rule 1.3** *Where a compound is ambiguous and is not clarified by the sentential context, the most typical meaning of the compound is favoured.*

Compound interpretation must sometimes rely on world knowledge. In these cases, the annotator will have to rely on his or her intuition. Querying Google for the most typical meaning would be a viable option, but would take too much time in the annotation process.

The compound *school book* is not clarified by a sentence such as *This is a school book*. In this case, book read IN school is the most typical interpretation. If the compound's ambiguity arises from the polysemy of a constituent, the same consideration applies. University can refer to an institution or its physical location, but in the case of *university degree* the institutional meaning must be correct as

locations cannot award degrees, and the compound is labelled ACTOR.
If the meaning of the compound is unclear, the appropriate tag is UNKNOWN.

**Rule 1.4** *There are number of special cases that would normally not appear in our training data. If they should be present, they are to be treated differently than other compounds, they will all be annotated REL.*

*- When a compound is used metaphorically, it will not be considered a regular compound and it should be labelled REL.*

For example: the compound *bird brain* is often used to refer to someone stupid, not to an actual bird's brain. Luckily, a lot of metaphorical compounds have such a typical meaning that they can be found in a dictionary and will therefore not be present in the annotation data.

*- Where a compound consisting of two common nouns is used as a proper noun, it will be discarded from our annotation. Also compounds that exist of one or more proper nouns, abbreviations or acronyms will be left out. All these special cases receive the REL tag.*

Many names, while constructed from two common nouns, do not seem to encode the same kind of semantics as non-name compounds, e.g. *Penguin Books, Sky Television, Dolphin Close, Coronation Street*. These names encode only a sense of non-specific association between the constituents. All compounds that are used as a proper noun will therefore be classified as REL, even those that could be classified otherwise. For example: the *Telecommunications Act*, *The Old Tea Shop*, *Castle Hill*. The task of identifying these proper noun compounds should be passed on to a named entity recognition (NER) module.

**Rule 1.5** *Where there is a characteristic situation or event that characterizes the semantic relation between the constituents, it is necessary to identify which constituents of the compound are participants and which roles they play. Whether such a situation exists for a given compound, and the roles played by its constituents in the situation, will determine which relation labels are available.*

Participants take on roles that can be described as Agent, Instrument, Object or Result:

- **Agent** The instigator of the event, the primary source of energy
- **Instrument** An intermediate entity that is used/acted on by the Agent and in turn exerts force on or changes the Object; more generally, an item which is used to facilitate the event but which is not the Object
- **Object** The entity on which a force is applied or which is changed by the event and which does

not exert force on any participant other than the Result. Recipients (e.g. of money or gifts, but not outcomes) also count as Objects.

- **Result** An entity which was not present before and comes into being through the event

For example, the meaning of *cheese knife* seems to involve an event of cutting, in which cheese and knife take object and instrument roles respectively. Similarly, *taxi driver* evokes an event of driving and *gevangenisbewaker* (prison guard) evokes an event of guarding. The INST and ACTOR relations apply only where such a situation or event is present and where the compound identifies its participant(s). The application of HAVE assumes that the most salient aspect of the underlying situation is possession. It is not strictly necessary to identify the precise nature of the situation or event, only to identify the general roles played by the participants.

Some role-tagged examples: *cheese*$_O$ *knife*$_I$, *taxi*$_O$ *driver*$_A$, *sneezing*$_R$ *powder*$_I$, *gevangenis*$_O$*bewaker*$_A$. It follows from the role descriptions that locations and topics do not count as participants – compounds encoding such roles receive IN and ABOUT labels instead of the ACTOR and INST labels reserved for participants.

The participant role types are listed in order of descending agentivity. We thus have an agentivity hierarchy Agent>Instrument>Object>Result[5]. This ordering plays an important role in distinguishing ACTOR compounds from INST compounds (see Rules 2.1.4 and 2.1.5). It is not necessary to annotate this information, and it is not always necessary to identify the exact participant role of a constituent, so long as the hierarchical order of the constituents can be identified. Identifying participants is only needed to distinguish between relations (ACTOR vs INST) and directionalities (see the discussion under Rule 2.1.5.2).

## 2. Semantic Relations

### 2.1 Main Relations

#### 2.1.1 BE

**Rule 2.1.1.1** *X is N1 and X is N2.*

For example:

[5] This agentivity hierarchy was informed by the semantic roles hierarchy in Talmy, 2000.
Talmy, L. (2000). 'The semantics of causation'. In: *Toward a Cognitive Semantics, Volume 1: Concept Structuring Systems.* Cambridge, MA: MIT Press.

English: *woman driver, elm tree, distillation process, human being.*
Dutch: *geluidhinder, rundsvlees, bombrief, puntkomma, gastarbeider, getuige-deskundige.*

This rule does not admit sequences such as *deputy chairman*, *fellow man*, *chief executive* or *hoofdverantwoordelijke*, where it is not correct to state that an [N1 N2] is an N1 (a chief executive is not a chief). Such sequences are not to be considered compounds, and their modifiers are to be considered (mistagged) adjectives – see Rule 2.2.1.1.

**Rule 2.1.1.2** *N2 is a form/shape taken by the substance N1.*

For example:
English: *stone obelisk, chalk circle, plastic box, steel knife*.
Dutch: *gummiband*, *betonsteen, staalkabel.*

This rule is not very productive in Dutch since substances are most often written as adjectives, e.g. *plastieken doos, stalen mes*.

**Rule 2.1.1.3** *N2 is ascribed significant properties of N1 without the ascription of identity. The compound roughly denotes "an N2 like N1".*

For example:
English: *father figure, angler fish, chain reaction, pie chart*.
Dutch: *hagelpatroon, rondegang, manwijf, mensaap.*

2.1.2 HAVE

**Rule 2.1.2.1** *N1/N2 owns N2/N1 or has exclusive rights or the exclusive ability to access or to use N2/N1 or has a one-to-one possessive association with N2/N1.*

For example:
(The numbers after the examples refer to the direction of the semantic relation.)
English: *army base(1), customer account(1), government power(1)*.
Dutch: *straatnaam(1), koningsdochter(1).*

The term one-to-one possessive association is intended to cover cases where it seems strange to speak of ownership, for example in the case of inanimate objects (*street name, planet atmosphere*).

**Rule 2.1.2.2** *N1/N2 is a physical condition, a mental state or a mentally salient entity experienced by N2/N1.*

For example:
English: *polio sufferer(1), cat instinct(2), student problem(2), union concern(2).*
Dutch: *lepralijder(1), studentenprobleem(2).*

**Rule 2.1.2.3** *N1/N2 has the property denoted by N2/N1.*

For example:
English: *water volume(1), human kindness(1).*
Dutch: *productietijd(1).*

A "property" is something that is not an entity or a substance but which an entity/substance can be described as having. *Redness, temperature, dignity, legibility* are all examples of properties.

**Rule 2.1.2.4** *N1/N2 has N2/N1 as a part or constituent.*

For example:
English: *car door(1), motor boat(2), cat fur(1), chicken curry(2), pie ingredient(1), tree sap(1).*
Dutch: *houtweefsel(1)*, *bladzijde(1), moutjenever(2), hamersteel(1), grafzerk(1), tafelblad(1).*

The test for the presence of a part-whole relation is whether it seems natural and accurate in the context to say *The N1/N2 has/have N2/N1* and *The N1/N2 is/are part of N2/N1*. Furthermore, substances which play a functional role in a biological organism are classed as parts: *human blood, tree sap, whale blubber*. This is the case even when the substance has been extracted, as in *olive oil*. A part is often located in its whole, but in these cases the part-whole relation is to be considered as prior to the co-location, and HAVE is preferred to IN. Complications arise with cases such as *sea chemical*, where both HAVE and IN seem acceptable. One principle that can be used tests whether the candidate part is readily separated (perceptually or physically) from the candidate whole. Chemicals in *sea water* (HAVE) are not typically separable in this way and can be viewed as parts of a whole. On the other hand, a *sea stone* or a *sea (oil) slick* are perceptually distinct and physically separable from the sea and are therefore IN.

**Rule 2.1.2.5** *N1/N2 is a group/society/set/collection of entities N2/N1*

For example:

English: *stamp collection(2), character set(2), lecture series(2), series lecture(1), committee member(1), infantry soldier(1).*
Dutch: *postzegelverzameling(2), schoenenhoop(2), groepslid(1).*

### 2.1.3 IN

In the following rules, an opposition is drawn between events/activities and objects. The class of events includes temporal entities such as times and durations. Objects are perceived as non-temporal and may be participants in an event (the term participant is used as defined under Rule 1.5). To assign the correct rule, the annotator must decide whether the located thing is an event or an object, and whether the location is temporal or spatial. Events may also sometimes be participants – in the sense of Rule 1.5 and in these cases the rules dealing with objects and participants will apply – a *nursing college* is a college where nursing is taught as a subject, but not necessarily one where the activity of nursing takes place, so Rule 2.1.3.1 applies. In contrast a *nursing home*, being a home where the event of nursing takes place, would come under Rule 2.1.3.2, analogous to *dining room*. Some nouns are polysemous and can refer to both objects (*play* as a written work, *harvest* as harvested crops) and events (*play* as performance, *harvest* as activity). The annotator must decide whether the temporal or physical aspect is primary in a given context.

**Rule 2.1.3.1** *N1/N2 is an object spatially located in or near N2/N1.*

For example:
English: *forest hut(2), shoe box(1), side street(2), top player(2), crossword page(1), hospital doctor(2), sweet shop(1).*
Dutch: *waterplant(2)*, *rivierleem(2), ziekenhuisbed(2), havenkantoor(2), kerkdief(2).*

Where the location is due to part-whole constituency or possession, HAVE is preferred (as in *car door*, *sea salt*). Source-denoting compounds such as *country boy* and *spring water* are classed as IN as the underlying relation is one of location at a (past) point in time.

**Rule 2.1.3.2** *N1/N2 is an event or activity spatially located in N2/N1.*

For example:
English: *dining room(1), hospital visit(2), sea farming(2), football stadium(1).*
Dutch: *biljartzaal(1), distributiecentrum(1), tuinfeest(2), zeeslag(2).*

**Rule 2.1.3.3** *N1/N2 is an object temporally located in or near N2/N1, or is a participant in an event/activity located there.*

For example:
English: *night watchman(2), coffee morning(1).*
Dutch: *nachtuil(2)*, *sterrennacht(1), lenteweertje(2), weekblad(2).*

**Rule 2.1.3.4** *N1/N2 is an event/activity temporally located in or near N2/N1.*

For example:
English: *future event(2), midnight mass(2)*.
Dutch: *avondfeest(2), nachtvoorstelling(2), jaarvergadering(2).*

2.1.4 ACTOR

The distinction between ACTOR and INST is based on sentience. Only certain classes of entities may be actors:

1. Sentient animate lifeforms: membership of the animal kingdom (regnum animalia) is a sufficient condition. Bacteria and viruses are not sentient enough (flu virus is annotated INST).
2. Organisations or groups of people: for example *finance committee, consultancy firm, manufacturing company, council employee*. Some words referring to institutions are polysemous in that they can denote its physical aspect or its social/organisational aspect – university often denotes a physical location, but in the compounds *university degree* and *university decision* it is functioning as an organisation and count as agents (granting a degree and making a decision are actions only humans or organisations can carry out). On the other hand, in *research university* it is not clear whether we have a university that does research (agentive) or a university in which research is done (non-agentive). In such cases, the physical denotation should be considered the primary meaning of the word, and the organisational denotation is derived through metonymy – the non-agentive interpretation of these compounds is favoured unless the underlying event requires the institution to act as an agent. Such events often involve the institution acting as a legal entity. Hence *university degree* (degree awarded by a university), *school decision* (decision made by a school), *shop employee* (employee employed by a shop) are ACTOR; *research university, community school, school homework* and *sweet shop* are IN.

A compound can be labelled ACTOR only if the underlying semantic relation involves a characteristic situation or event. In the following definitions, the term participant is used in the sense of Rule 1.5.

**Rule 2.1.4.1** *N1/N2 is a sentient participant in the event N2/N1.*

For example:
English: *student demonstration(1), government interference(1), infantry assault(1).*
Dutch: *burgeroorlog(1)*, *arbeidsvrouw(2), aanslagpleger(2).*

That N2/N1 denote an event is not sufficient for this rule – it must be the characteristic event associated with the compound. Hence this rule would not apply to a *singing teacher*, as the characteristic event is teaching, not singing. Instead, Rule 2.1.4.2 would apply. As only one participant is mentioned in the current rule 2.1.4.1, there is no need to establish its degree of agentivity.

**Rule 2.1.4.2** *N1/N2 is a sentient participant in an event in which N2/N1 is also a participant, and N1/N2 is more agentive than N2/N1.*

For example:
English: *honey bee(2), bee honey(1), company president(2), history professor(2), taxi driver(2), student nominee(1).*
Dutch: *aasdier(2), hartendief(2).*

Relative agentivity is determined by the hierarchy given under Rule 1.5. The underlying event cannot be one of possession (*car owner* = HAVE) or location (*city inhabitant* = IN). Profession-denoting compounds often have a modifier which is a location – *street cleaner, school principal, restaurant waitress, school teacher.* A distinction can be drawn between those where the profession involves managing or changing the state of the location, i.e. the location is an object (*school principal, street cleaner* = ACTOR), and those where the profession simply involves work located there (*school teacher, restaurant waitress* = IN by Rule 2.1.3.1). Note that modifiers in *-ist* such as *expressionist*, *modernist*, *socialist*, *atheist* are treated as nouns, so that an *expressionist poem* is analysed as a poem such as an expressionist would characteristically write.

### 2.1.5 INST

The name INST(rument) is used to distinguish this category from ACTOR, though the scope of the category is far broader than traditional definitions of instrumentality. Again, the term participant is used in the sense of Rule 1.5.

**Rule 2.1.5.1** *N1/N2 is a participant in an activity or event N2/N1, and N1/N2 is not an ACTOR.*

For example:
English: *skimming stone(2), gun attack(1), gas explosion(1), combustion engine(2), drug trafficking(1), rugby tactics(2), machine translation(1).*
Dutch: *smaakbederf(1)*, *zaadhandel(1)*, *leengoed*(2).

Compounds identifying the location of an event (such as *street demonstration*) should be labelled IN by Rule 2.1.3.2 or 2.1.3.4, and compounds identifying the focus of or general motivation for a human activity or mental process (such as *crime investigation*), but not its direct cause, should be labelled ABOUT by Rule 2.1.6.3.
As only one participant is mentioned, there is no need to establish its degree of agentivity.

**Rule 2.1.5.2** *The compound is associated with a characteristic event in which N1/N2 and N2/N1 are participants, N1/N2 is more agentive than N2/N1, and N1/N2 is not an ACTOR.*

For example:
English: *rice cooker(2)*, *tear gas(2)*, *blaze victim(1).*
Dutch: *cadeaubon(2), worstmachine(2).*

The directionality of the relation is determined by the more agentive participant in the hierarchy given in Rule 1.5: *cheese*$_\mathrm{O}$ *knife*$_\mathrm{I}$ (INST2), *wine*$_\mathrm{O}$ *vinegar*$_\mathrm{R}$ (INST1), *wind*$_\mathrm{A}$ *damage*$_\mathrm{R}$ (INST1), *human*$_\mathrm{O}$ *virus*$_\mathrm{A}$ (INST1). Sometimes it may be difficult to distinguish Agents from Instruments (*gun wound*) or Objects from Results (*blaze victim*) – this is not important so long as it is possible to identify which participant is more agentive.
In some cases, it may not be clear what the exact underlying event is, but the more agentive participant may still be identified – a *transport system* is a system that in some way provides or manages transport, but it is nonetheless clear that the appropriate label is INST2. In other cases, where both participants affect each other, it may be less clear which is more agentive – *motor oil* can be construed as oil that lubricates/enables the function of the engine or as oil the engine uses. Likewise *petrol motor, computer software, electron microscope*. At least where the relation is between a system or machine and some entity it uses to perform its function, the former should be chosen as more agentive. Hence *motor oil* is INST1, *petrol motor* is INST2, and so on.
As in Rule 2.1.5.1, where one of the constituents is the location of the associated event, then IN is the appropriate label by Rule 2.1.3.1 or 2.1.3.3. If the more agentive participant meets the criteria for ACTOR status (2.1.4), then that label should be applied instead. If the interaction between the constituents is due to one being a part of the other (as in car engine), HAVE is the appropriate label by

Rule 2.1.2.4. A border with ABOUT must be drawn in the case of psychological states and human activities whose cause or focus is N1. As described further under Rules 2.1.6.3, the criterion adopted is based on whether there is a direct causal link between N1 and N2 in the underlying event – a bomb can by itself cause *bomb terror* (INST1), but a *spider phobia* is not a reaction to any particular spider and is classed as ABOUT.

2.1.6 ABOUT

**Rule 2.1.6.1** *N1/N2's descriptive, significative or propositional content relates to N2/N1.*

For example:
English: *fairy tale(2), flower picture(2), tax law(2), exclamation mark(2), film character(2), life principles(2), sitcom family(1).*
Dutch: *vakjargon(2), contactstoornis(2), praktijktheorie(2), vakdeskundigheid(2).*

In English, a lot of speech acts belong to this category. Direction 2 is a lot more prominent with this rule. Properties and attributes that seem to have a descriptive or subjective nature are still to be labelled HAVE by Rule 2.1.2.3 – *street name* and *music loudness* are HAVE1.

**Rule 2.1.6.2** *N1/N2 is a collection of items whose descriptive, significative or propositional content relates to N2/N1 or an event that describes or conveys information about N2/N1.*

For example:
English: *history exhibition(2), war archive(2), science lesson(2).*
Dutch: *tijdreeks(2), muziekbibliotheek(2).*

**Rule 2.1.6.3** *N1/N2 is a mental process or mental activity focused on N2/N1, or an activity resulting from such.*

For example:
English: *crime investigation(2), science research(2), research topic(1), exercise obsession(2), election campaign(2), football violence(2), holiday plan(2).*
Dutch: *darmonderzoek(2),*

In the case of activities, N1/N2 cannot belong to any of the participant categories given under Rule 1.5; rather it is the topic of or motivation for N2/N1. The sense of causation in, for example, *oil dispute* is not direct enough to admit an INST classification – the state of the oil supply will not lead to an oil dispute without the involved parties taking salient enabling action. In the case of emotions, there is also a risk of overlapping with INST; *bomb terror* is INST and *bomb dislike* is classed as ABOUT, but examples such as *bomb fear* are less clearcut. A line can be drawn whereby immediate emotional reactions to a stimulus are annotated INST, but more permanent dispositions are ABOUT. In the case of bomb fear, the relation must be identified from context. Problems (*debt problem*) and crises (*oil crisis*) also belong to this category, as they are created by mental processes.

**Rule 2.1.6.4** *N1/N2 is an amount of money or some other commodity given in exchange for N2/N1 or to satisfy a debt arising from N2/N1.*

For example:
English: *share price(2), printing charge(2), income tax(2).*
Dutch: *olieprijs*(2), *loonarbeid*(1), *gokbedrag*(2).

N2/N1 is not the giver or recipient of N1/N2 – an *agency fee* would be INST under the interpretation fee$_I$ paid to an agency$_O$ – but the thing exchanged or the reason for the transaction.

2.1.7 REL

**Rule 2.1.7.1** *The relation between N1 and N2 is not described by any of the above relations but seems to be produced by a productive pattern.*

For example:
English: *Baker Street, sodium chloride,*
Dutch: *Vaarttheater, Plataanlei, waterstofcarbonaat, adjudant-onderofficier.*

A compound can be associated with a productive pattern if it displays substitutability. If both of the constituents can be replaced by an open or large set of other words to produce a compound encoding the same semantic relation, then a REL annotation is admissible. For example, the compound *reading skill* (in the sense of degree of skill at reading) is not covered by any of the foregoing categories, but the semantic relation of the compound (something like ABILITY) is the same as that in *football skill*, *reading ability* and *learning capacity*. This contrasts with an idiosyncratic lexicalised compound such as *home secretary* (= LEX), where the only opportunities for substitution come from a restricted class and most substitutions with similar words will not yield the same semantic relation. Another class of compounds that should be labelled REL are names of chemical compounds such as *carbon dioxide* and *sodium carbonate*, as they are formed according to productive patterns. There are also several special cases that receive the REL tag. Take a look at Rule 1.4 for the descriptions.

### 2.1.8 LEX

**Rule 2.1.8.1** *The meaning of the compound is not described by any of the above relations and it does not seem to be produced by a productive pattern.*

For example:
English: *turf accountant, monkey business*.
Dutch: *loftrompet, prins-gemaal,*

These are noncompositional in the sense that their meanings must be learned on a case-by-case basis and cannot be identified through knowledge of other compounds. This is because they do not have the property of substitutability - the hypothetical compounds *horse business* or *monkey activity* are unlikely to have a similar meaning to *monkey business*. LEX also applies where a single constituent has been idiosyncratically lexicalised as a modifier or head such as *X secretary* meaning ‘minister responsible for X’.

### 2.1.9 UNKNOWN

**Rule 2.1.9.1** *The meaning of the compound is too unclear to classify.*

Some compounds are simply uninterpretable, even in context. This label should be avoided as much as possible but is sometimes unavoidable.

## 2.2 Noncompounds

### 2.2.1 MISTAG

**Rule 2.2.1.1** *One or both of N1 and N2 have been mistagged and should not be counted as (a) common noun(s).*

For example:
English: *fruity bouquet* (N1 is an adjective), *London town* (N1 is a proper noun).
Dutch: *Juratijdperk* (N1 is a proper noun), *voortuin* (N1 is a preposition), *hoofdbewaker* (N1 is adjective-like).

In the case of *blazing fire*, N1 is a verb, so this is also a case of mistagging; in superficially similar cases such as *dancing teacher* or *swimming pool*, however, the -ing form can and should be treated as a noun. The annotator must decide which analysis is correct in each case – a *dancing teacher* might be a teacher who is dancing (MISTAG) in one context, but a teacher who teaches dancing (ACTOR) in another context. Certain modifiers might be argued to be nouns but for the purposes of annotation are stipulated to be adjectives. Where one of *assistant, key, favourite, deputy, head, chief* or *fellow* appears as the modifier of a compound in the data, it is to be considered mistagged. This only applies when these modifiers are used in adjective-like senses – *key chain* or *head louse* are clearly valid compounds and should be annotated as such.

### 2.2.2 NONCOMPOUND

**Rule 2.2.2.1** The *extracted sequence, while correctly tagged, is not a 2-noun compound.*

There are various reasons why two adjacent nouns may not constitute a compound:

1. An adjacent word should have been tagged as a noun, but was not.
2. The modifier is itself modified by an adjacent word, corresponding to a bracketing [[X N1] N2]. For example: [[*real tennis*] *club*], [[*Liberal Democrat*] *candidate*], [[*five dollar*] *bill*]. However compounds with conjoined modifiers such as *land and sea warfare* and *fruit and vegetable seller* can be treated as valid compounds so long as the conjunction is elliptical (*land and sea warfare* has the same meaning as *land warfare* and *sea warfare*). Not all conjoined modifiers satisfy this condition – a *salt and pepper beard* does not mean a beard which is a *salt beard* and a *pepper beard*, and the sequence *pepper beard* is a NONCOMPOUND.
3. The two words are adjacent for other reasons. For example: ‘the *question politicians* need to answer’, structureless lists of words.
4. The modifier is not found as a noun on its own, because it would not appear in the dictionary. For example: *multiparty election*, *smalltown atmosphere*.