*Natural Language Engineering* **20** (4): 537–555. © Cambridge University Press 2013 
doi:10.1017/S1351324913000053

# *Lexical acquisition and semantic space models: Learning the semantics of unknown words*

KOSTADIN CHOLAKOV

*University of Groningen, Oude Kijk in 't Jatstraat 26, 9712EK Groningen, The Netherlands*
*e-mail*: k.cholakov@rug.nl

(*Received 19 December 2011; revised 5 February 2013; accepted 11 February 2013; first published online 5 March 2013*)

**Abstract**

In recent studies it has been shown that syntax-based semantic space models outperform models in which the context is represented as a bag-of-words in several semantic analysis tasks. This has been generally attributed to the fact that syntax-based models employ corpora that are syntactically annotated by a parser and a computational grammar. However, if the corpora processed contain words which are unknown to the parser and the grammar, a syntax-based model may lose its advantage since the syntactic properties of such words are unavailable. On the other hand, bag-of-words models do not face this issue since they operate on raw, non-annotated corpora and are thus more robust. In this paper, we compare the performance of syntax-based and bag-of-words models when applied to the task of learning the semantics of unknown words. In our experiments, unknown words are considered the words which are not known to the Alpino parser and grammar of Dutch. In our study, the semantics of an unknown word is defined by finding its most similar word in CORNETTO, a Dutch lexico-semantic hierarchy. We show that for unknown words the syntax-based model performs worse than the bag-of-words approach. Furthermore, we show that if we first learn the syntactic properties of unknown words by an appropriate lexical acquisition method, then in fact the syntax-based model does outperform the bag-of-words approach. The conclusion we draw is that, for words unknown to a given grammar, a bag-of-words model is more robust than a syntax-based model. However, the combination of lexical acquisition and syntax-based semantic models is best suited for learning the semantics of unknown words.

## 1 Introduction

Vector-based semantic space models have been successfully employed to represent lexical meaning in a variety of Natural Language Processing (NLP) tasks, such as information retrieval (Salton, Wong and Yang 1975), word sense discrimination (Schütze 1998; McCarthy *et al.* 2004), concept categorization (Almuhareb and Poesio 2004), verb selectional preferences (Erk 2007), and automatic thesaurus extraction (Grefenstette 1994; Lin 1998a; Curran and Moens 2002; Rapp 2004). The research carried out in recent years has been concentrated mainly on two types of semantic space models, namely bag-of-words and syntax-based models.[1]

[1] For a thorough survey of vector-based semantic space models, see Turney and Pantel (2010).

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

The intuition behind semantic space models is that similar words occur in similar contexts (Miller and Charles 1991). Accordingly, such models collect co-occurrence relations between a word and its context. In a bag-of words model, the context of a word is represented by words occurring within some distance of this word. The context in a syntax-based model is usually represented by triples collected from parsed corpora. Such a triple has the following elements: the target word, a second word which is linked to the target one by some syntactic relation, and the type of syntactic relation linking the two words. Let us consider the following sentence:

(1) The Simpsons drive a blue car.

In a bag-of-words model *drive* is the context of both *blue* and *car* because it appears close to those words. However, the model does not take into account the fact that, besides linear proximity, there is no real linguistic relation between *blue* and *drive*. A syntax-based model, on the other hand, will not consider *drive* as the context of *blue* but it will identify the *object* relation between *drive* and *car*. Moreover, the model will distinguish it as a different type of relation than the *modifier* relation between *blue* and *car*.

The richer type of information provided comes at a price. Compared to bag-of-words, syntax-based models are more complex and require more preliminary processing: parsing, identification and extraction of syntactic patterns. Such models heavily depend on the quality of the syntactic analyses delivered by a parser and the computational grammar this parser employs. Still, it has been shown (Lin 1998a; Van der Plas and Tiedemann 2006; Padó and Lapata 2007; Van de Cruys 2008; Rothenhäusler and Schütze 2009) that models which consider syntactic relations generally outperform bag-of-words models in a number of NLP and cognitive science tasks, such as synonymy detection, word sense disambiguation, word categorization, and semantic priming.

In this paper we compare the performance of these two types of semantic space models when applied to the task of learning the semantics of words which are unknown to a given computational grammar. Since no syntactic information for such words is available, the performance of a syntax-based model could suffer. A bag-of-words model, on the other hand, might perform better because it does not consider any syntactic information. In our study, we compute the semantics of a given unknown word by identifying its most similar word in CORNETTO (Horak, Vossen and Rambousek 2008), a Dutch lexico-semantic hierarchy similar to WordNet (Fellbaum 1998). To do so, in our experiments, we employ the Alpino grammar and dependency parser of Dutch (van Noord 2006).

The performance of three semantic space models is compared. The first one is syntax-based. We employ the lexical acquisition (LA) method that we presented in Cholakov and van Noord (2010). This is an automated method for acquiring lexical entries for the lexicon of a given grammar. Note that throughout the paper the term **lexical acquisition** is used in a 'parsing' context, i.e. this term refers to the acquisition of novel lexical entries for a lexical resource used for parsing. The method of Cholakov and van Noord (2010) is applied to learn lexical entries for a test set of words unknown to the Alpino grammar and parser. After extending the lexicon of Alpino with these entries, we parse large Dutch corpora (∼2 billion words). The syntax-based semantic model then employs dependency relations extracted from the parsed corpora as contexts for the unknown words.

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

The second model we build is another syntax-based one. It uses dependency relations from the same corpora but when parsed with the standard Alpino configuration. In this case, a built-in unknown word guesser is used to guess the syntactic properties of the unknown words from the test set. Finally, the third model we explore is a bag-of-words model.

The performance of all three models is evaluated automatically against the CORNETTO hierarchy. For a sample set of candidate words (nouns and verbs) randomly extracted from CORNETTO, the task of the models is to map each word in the test set to the candidate word which is most similar to it. Note that, for reasons which will be given in Section 4, adjectives are not included in the experiments. The bag-of-words model performs better than the syntax-based one which uses contexts from the corpora parsed with the default configuration of Alpino. The insufficient quality of the syntactic information provided by the parser's built-in guesser for the unknown words indeed affects the quality of this syntax-based model. However, the syntax-based model which employs our LA method achieves the best performance of all three models.

The conclusion we draw is that in the case of words with unknown syntactic properties, bag-of-words models are more robust than syntax-based ones but when the latter are combined with an appropriate LA technique, such as our method, the situation changes. The combination of LA and syntax-based semantic models achieves the best performance in learning the semantics of unknown words.

The remainder of the paper is organised as follows. Section 2 gives a brief introduction to bag-of-words and syntax-based semantic space models. Section 3 provides background information on LA and details about our LA method. Section 4 describes the CORNETTO lexico-semantic hierarchy, the evaluation framework, and the test set of unknown words we employ for the experiments presented in the paper. Section 5 shows the results for the three semantic space models explored here. Section 6 concludes the paper.

## 2 Semantic space models

As mentioned in Section 1, we discuss two kinds of semantic space models – bag-of-words and syntax-based models. Lowe (2001) defines a semantic space model as a quadruple $\langle A, B, S, M \rangle$. $B$ is a set of basic elements $(b_1 \ldots b_D)$ determining the dimensionality $D$ of the vector space. $B$ might be a set of documents, words, or dependency relations, depending on the context that is used. Usually, the dimensionality is restricted to some number $k$, for example the $k$ most frequent words in a corpus. The basic elements are often called **features**. Henceforth, we will use this term. $A$ is a lexical association function applied to the co-occurrence frequency of target word $t$ with a feature $b \in B$ so that each word is represented by a vector $\vec{v} = [A(b_1,t), A(b_2,t), \ldots, A(b_D,t)]$. $A$ may simply map to the raw frequencies of the target word but often a more advanced mapping is used. $S$ is a similarity measure that maps pairs of vectors onto a real number that represents semantic similarity. Finally, $M$ is an optional mathematical transformation that reduces the dimensionality of the semantic space. For example, singular value decomposition (SVD; Golub and Loan 1989; Berry, Dumais and O'Brien 1994) is commonly used for this purpose.

Let us give an example by constructing a bag-of-words semantic space for the set of target words $T = \{jongen, wijn\}$ based on the following single-sentence Dutch corpus:

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

| | mooie | jongen | drinkt | lekkere | wijn | buiten |
|---|---|---|---|---|---|---|
| jongen | 1 | 0 | 1 | 1 | 0 | 0 |
| wijn | 0 | 0 | 1 | 1 | 0 | 1 |

Fig. 1. Example of a word-based semantic space.

(2) De mooie jongen drinkt lekkere wijn buiten.
the nice boy drinks tasty wine outside
'The nice boy is drinking tasty wine outside.'

Often, stop words are ignored and so our set of features is $B = \{mooie, jongen, drinkt, lekkere, wijn, buiten\}$. $A$ is the identity function and we use a context window of two words either side of the target word. Each target word $t_i \in T$ is then represented as a vector, and the value of each vector element shows the number of occurrences of each feature $b_j \in B$ within the context window of target word $t_i$. Figure 1 shows the co-occurrence matrix containing the vectors for the two target words.

Padó and Lapata (2007) extend Lowe's framework so that it can handle the construction of semantic space models based on syntax (dependency parses). Dependency parses are interpreted as directed graph structures of nodes and labelled edges. A particular dependency path $\pi$ for a given target word can then be represented as an ordered set of tuples according to the dependency graph.

In this extended framework, a semantic space is a tuple $\langle B, T, M, S, A, cont, \mu, v \rangle$. $B$ is the set of basic elements, $T$ is the set of target words, $M$ is a matrix with size $|B| \times |T|$, $A$ is the lexical association function, and $S$ is the similarity measure. These parameters do not differ significantly from Lowe's model. The additional parameters used are the context selection function $cont : T \rightarrow 2^{\pi}$, the basic mapping function $\mu : \pi \rightarrow B$, and the path value function $v : \pi \rightarrow \mathbb{R}$.

The context selection function *cont* provides for the selection of certain paths in the graph that contribute to the context of a particular target word. This function allows us to select only paths with a particular length, or paths that are labelled with a particular dependency relation. The basic mapping function $\mu$ maps paths onto basic elements. In this way, the dependency paths are decoupled from their representation in the final semantic space. Such decoupling allows, for example, to use words instead of syntactic features in the final representation. The path value function $v$ is used to assign weights to particular paths. For example, longer dependency paths might be given less weight. Another possibility is to give more weight to particular dependency relations.

Having this extended framework for constructing semantic space models, we parse the sentence in 2 with the Dutch Alpino dependency parser. The obtained dependency triples involving the two target words, *jongen* and *wijn*, are given in Figure 2. These dependency relations are then used as the set of features $B$ in syntax-based semantic space models, i.e. as the context of the target words. The values of the parameters for the syntax-based models we employ are discussed in Section 5, which presents the results of our experiments.

Once the co-occurrence matrix is constructed, a variety of similarity measures can be applied to calculate the similarity $S$ between the vectors $\vec{v}, \vec{w}$ that represent the two target

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

⟨ jongen mod mooie ⟩
⟨ jongen det de ⟩
⟨ drinkt su jongen ⟩
⟨ drinkt obj wijn ⟩
⟨ wijn mod lekkere ⟩

Fig. 2. Extracted dependency relations.

words, the cosine being the most popular:

$$cos(\vec{v},\vec{w}) = \frac{\sum_{i=1}^{n} v_i w_i}{\sqrt{\sum_{i=1}^{n} v_i^2}\sqrt{\sum_{i=1}^{n} w_i^2}} \tag{1}$$

Often, the lexical association function $A$ is modified so that each feature receives a value in the matrix according to its importance. Several weighting schemes have been used for this purpose, one of them being pointwise mutual information (PMI). It was first proposed by Church and Hanks (1990), and is based on the information–theoretic notion of mutual information (MI). Mutual information measures the mutual dependence between two random variables $X$ and $Y$. It is defined as

$$I(X,Y) = \sum_x \sum_y p(x,y) log \frac{p(x,y)}{p(x)p(y)} \tag{2}$$

PMI, the mutual information for particular events, is defined as

$$I(i,j) = log \frac{p(i,j)}{p(i)p(j)} \tag{3}$$

Intuitively, PMI indicates how much information a particular feature contains about a target word (and *vice versa*). Throughout the paper, we apply the cosine similarity measure and use PMI to weight the features in all semantic space models.

## 3 Lexical acquisition

### *3.1 Background*

Lexical acquisition can be defined as the automatic extension of the lexicon of a given grammar with new lexical entries. When learning a lexical entry for a given unknown word, LA techniques usually take into account different types of features of that word which vary in the complexity of the linguistic information they designate – affixes, part of speech (POS) tags, etc. This makes LA methods complex and time-consuming. That is why, they are often applied off-line. LA methods are usually evaluated in terms of **type precision** and **type recall**. For a given LA method, type precision indicates the proportion of correctly predicted lexical entries, and type recall indicates how many of the correct lexical entries for a given word are actually found.

Various LA techniques for computational grammars have been proposed in the past. Erbach (1990), Barg and Walther (1998), and Fouvry (2003) follow a unification-based approach. The basic idea is to assign to unknown words such lexical entries that have fewer constraints, i.e. entries where some of the linguistic information is missing, to allow the grammar to parse the whole sentence. The full lexical entries are generated afterwards

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

by collecting the information from the full parse. This approach is a way of letting the grammar decide what is the best entry for a given unknown word. However, the generated lexical entries might be too general or too specific which makes the application of the aforementioned methods to large grammars problematic. Furthermore, no meaningful evaluation has been provided on how these methods affect parsing accuracy.

Baldwin (2005) uses resources outside the grammar to obtain linguistic information about an unknown word. The method employs a set of binary classifiers with features extracted from POS taggers, chunkers, ontologies, and other linguistic resources to learn lexical entries for a large-scale grammar of English (ERG; Copestake and Flickinger 2000). Since the classifiers learn full-fledged lexical entries, this approach does not have the shortcomings of the unification-based methods. The highest result (about 65 percent F-measure) is achieved by using features from a chunker. However, no evaluation is done on how the method affects the parsing accuracy of the ERG for sentences containing unknown words.

Zhang and Kordoni (2006) and Cholakov, Kordoni and Zhang (2008) also treat LA as a classification task but, similarly to the unification-based approaches, these methods involve the grammar in the LA process. A maximum entropy (ME)-based classifier is trained with features, some of which come from the grammar. The methods are used to acquire new lexical entries for ERG and a Head-driven Phrase Structure Grammar (HPSG) of German (GG; Crysmann 2003). The features are obtained from the annotated treebanks associated with both grammars. However, when the methods are applied to open-text unannotated data, the grammar features are replaced with POS tags. Therefore, the grammar is no longer directly involved in the LA process which affects the quality of the predictions. Evaluation on sentences containing unknown words shows improvement in parsing coverage when LA is employed but the parsing accuracy decreases.

The LA method we describe in Cholakov and van Noord (2010) develops further the ideas presented in the previous work described above. We also treat LA as a classification problem and, similar to the methods applied to the ERG and GG, recognize the importance of the grammar involvement in the LA process. However, our method has three main advantages that distinguish it from the methods discussed above.

First, it enables the acquisition of the whole paradigm of the unknown word, while the other approaches are only concerned with the particular word form encountered in the data subject to LA. Baldwin (2005) has demonstrated the importance of linguistic information obtained from resources outside the grammar and we see the paradigm of the word as one such resource. It provides us with a valuable source of morphological information. If we were to deal with only one form of the unknown word, this information would not be accessible.

Second, we take into account different contexts in which morphological forms of the unknown word occur. This increases the amount and diversity of the linguistic data used by our method. Cases where this is particularly important include morphologically ambiguous words and verbs that subcategorise for various types of syntactic arguments.

The third advantage is the employment of a full-blown syntactic parser and the grammar for which we learn lexical entries to analyse the contexts of the unknown word. This allows for the efficient incorporation of the syntactic constraints these contexts impose on the unknown word into the LA process. Thus, the grammar can now be directly involved in the acquisition of lexical entries.

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

Table 1. *Features for inspraakprocedures (consultation procedures)*

| Features |
|---|
| **(i)** i, in, ins, insp |
| **(ii)** s, es, res, ures |
| **(iii)** particle_yes #in this case *in* |
| **(iv)** hyphen_no |
| **(v)** noun⟨de,pl⟩ |
| **(vi)** noun(de,count,pl), tmp_noun(de,count,sg) |
| **(vii)** noun(de), noun(count), noun(pl), tmp_noun(de) tmp_noun(count), tmp_noun(sg) |

### *3.2 Our lexical acquisition method*

The method was originally developed for the Dutch Alpino grammar and parser, but we have demonstrated (Cholakov *et al.* 2011) that it can be easily used for other languages and systems. Here we give a short description of Alpino and an example of acquiring a lexical entry for an unknown Dutch word.

The Alpino parser is based on a large stochastic attribute value grammar. The grammar takes a 'constructional' approach, with rich lexical representations stored in the lexicon and a large number of detailed, construction-specific rules (about 800). Currently, the lexicon contains over 100,000 lexical entries and a list of about 200,000 named entities. Each word is assigned one or more lexical types. For example, the verb *boekt* (to book) is assigned two lexical types – *verb(hebben,sg3,intransitive)* and *verb(hebben,sg3,transitive)* – because it can be used either transitively or intransitively. The other type features indicate that it is a present third person singular verb and forms perfect tense with the auxiliary verb *hebben*.

The goal of the method is to assign correct lexical type(s) to a given unknown word. It takes into account only open-class lexical types: nouns, adjectives, and verbs. We assume that adverbs represent a closed class in Dutch. In addition, adjectives can be used as adverbs in Dutch. The types considered in the learning process are called **universal types**.

For a given word, an ME-based classifier[2] takes various morphological and syntactic features as input and outputs a ranked list of lexical types. The probability of a lexical type $t$, given an unknown word and its context $c$ is:

$$p(t|c) = \frac{exp(\sum_i \Theta_i f_i(t,c))}{\sum_{t' \in T} exp(\sum_i \Theta_i f_i(t',c))} \tag{4}$$

where $f_i(t,c)$ may encode arbitrary characteristics of the context and $< \Theta_1, \Theta_2, ... >$ is a weighting parameter that maximises the entropy and can be evaluated by maximising the pseudo-likelihood on a training corpus (Malouf 2002).

Table 1 shows the features for the noun *inspraakprocedures* (consultation procedures). Rows **(i)** and **(ii)** contain affix features derived from the word. The two features in rows **(iii)** and **(iv)** indicate whether the word starts with a separable particle and whether it contains a hyphen, respectively.

[2] TADM; http://tadm.sourceforge.net/

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

Further, the method we describe in Cholakov and van Noord (2009) is applied to generate the paradigm(s) of each word in question. This method uses a combination of finite state morphology and web search heuristics to generate possible paradigm(s) for a given word. A single paradigm is generated for *inspraakprocedures*, indicating that this word is a plural noun which takes the *de* definite article. This information is explicitly used as a feature in the classifier, which is shown in row **(v)** of Table 1.

Next, syntactic features for *inspraakprocedures* are obtained by extracting from a large corpus a number of sentences which the word occurs in. In our experiments the sentences are extracted from the Dutch Twente newspaper corpus (approximately 500 million words; Ordelman 2002). The sentences are parsed with a different 'mode' of Alpino where *inspraakprocedures* is assigned all universal types, i.e. the word is treated as being maximally ambiguous. For each sentence only that parse is preserved which is considered to be the best by the Alpino statistical disambiguation model. Then the lexical type that has been assigned to *inspraakprocedures* in this parse is stored. When all sentences have been parsed, a list can be drawn up with the types that have been used and their frequency:

(3) noun(de,count,pl) 78
tmp_noun(de,count,sg) 7
tmp_noun(het,count,pl) 6
proper_name(pl,'PER') 5
proper_name(pl,'ORG') 3
verb(hebben,pl,vp) 1

The lexical types assigned in at least 80 percent of the parses are used as features in the classifier.[3] These are the two features in row **(vi)**. Further, as shown in row **(vii)**, each attribute of the considered types is also taken as a separate feature.

The ME-based classifier described above must first be trained on a set of unknown words. For this purpose, in Cholakov and van Noord (2010), 2,000 words were temporarily removed from the Alpino lexicon, i.e. these words were made unknown to the grammar. The lexical entries which those words had in the lexicon were preserved and used as a gold standard in the training procedure. Once the classifier was trained, 500 more words were removed from the lexicon. Those words served as a test set. All words employed for training and testing the classifier had been randomly selected from a list of 4,537 words (out of the total 100,000 words in the lexicon). This list contained all lexicon words which had between fifty and 100 occurrences in a sample newspaper corpus consisting of 16 million words. It is low frequency words which are typically unknown to the grammar. Therefore, the manner in which the words employed for training and testing the classifier were selected aimed at simulating the behaviour of unknown words. Experiments with a minimum lower than fifty occurrences had shown that this was a reasonable threshold to filter out 'noise' such as typos, tokenization errors, etc.

The LA method was applied to assign lexical types to the 500 test words. We had preserved the original lexical types which the test words had in the Alpino lexicon and used those types as a gold standard for evaluation. The LA method achieved a type precision

[3] The 80 percent threshold is an empirical setting.

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

of 89.08 percent and the type recall was 80.52 percent, resulting in an F-measure (the harmonic mean of precision and recall) of 84.58 percent. Experiments with a set of 300 sentences containing real unknown words were also conducted. Those experiments showed that when LA is employed, the parsing accuracy of Alpino for those sentences improved significantly compared with the accuracy of the standard Alpino configuration.

## 4 Experimental setup

In this section, we describe the setup for the experiments we present and the resources used in them.

### *4.1 Lexico-semantic hierarchy*

The semantic space models we discuss in this paper are automatically evaluated by comparing them to the CORNETTO lexico-semantic hierarchy for Dutch. The intuition behind this comparison is that if two words are close to each other in the hierarchy, they are semantically similar. CORNETTO is a further development of the Dutch part of EuroWordNet (Vossen 1998). It was created semi-automatically by merging existing lexical resources for Dutch and manually adding lexico-semantic information and relations between the entries in the hierarchy. CORNETTO contains hierarchically structured IS-A relations. There are about 70,000 synsets which altogether account for about 120,000 words. Approximately 53,000 of those synsets denote nouns and another 9,000 denote verbs.

Figure 3 shows a sample extract of the CORNETTO hierarchy.[4] The synset on the top of the hierarchy contains only one word – *iets* (entity). As the highest point in the hierarchy, this synset ensures the existence of a path between any two synsets. However, it is important to note that CORNETTO is constructed semi-automatically and it is still in development. There are over 4,000 synsets which are not linked to the top of the hierarchy or to any other synset.

### *4.2 Evaluation framework*

A number of measures have been developed to calculate the similarity between two words in WordNet-like hierarchies. By looking at this distance, we can determine how similar these words are. In this paper, we use two similarity measures – the Wu & Palmer measure, which is based on path length, and the Lin information–theoretic measure.

Wu and Palmer (1994) propose a measure to compute the similarity between two words by using the path length between synsets $s_1$ and $s_2$ containing those words:

$$sim_{wp}(s_1, s_2) = \frac{2C}{A + B + 2C} \tag{5}$$

where $A$ ($B$) is the length of the path between $s_1$ ($s_2$) and the lowest common ancestor $lca(s_1, s_2)$ of $s_1$ and $s_2$. The path length is measured by counting the edges between the nodes in the CORNETTO hierarchy. $C$ is the length of the path from the global root of the hierarchy (in this case '*iets:noun:2*') to the *lca*.

[4] For reasons of space and simplicity, the nodes between 'iets:noun:2' and 'dier:noun:1' are represented by '...' in the node immediately below the top of the hierarchy.

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

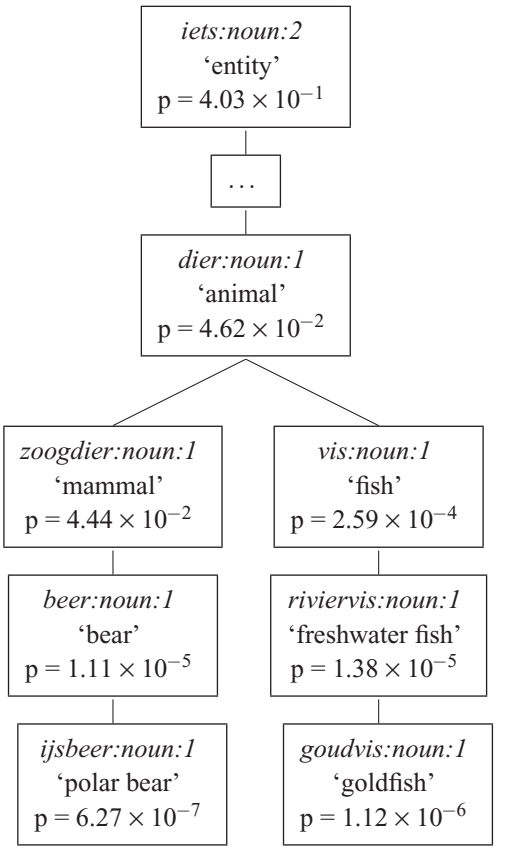



Fig. 3. CORNETTO extract.

Note that $A + B$ denotes the shortest path length between $s_1$ and $s_2$, and $C$ is the global depth of this path in the hierarchy. If a particular path has a large depth, it is given more weight compared to a path that is high up in the hierarchy. The intuition behind this is that the further down a synset, the more specific the concepts it denotes. Therefore, if the *lca* of two synsets is low in the hierarchy, they are likely to share many characteristics (i.e. to be more similar) as opposed to synsets whose *lca* is an abstract synset close to the top of the hierarchy.

The Wu & Palmer measure models this intuition by counting the edges between the nodes. However, the structure of the lexico-semantic hierarchies varies and is often unreliable. A way around this issue is to associate probabilities with the synsets in the hierarchy. If $p(s)$ is the probability of encountering a word belonging to synset $s$, we can calculate the information content (IC) of $s$ as the self-information of $p(s)$:

$$I(p(s)) = -\log p(s) \tag{6}$$

The probabilities are estimated by counting each word in a large corpus as an occurrence of the synset it belongs to, as well as an occurrence of all ancestors of this synset. The

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

probability of a synset $s$ then amounts to its relative frequency:

$$p(s) = \frac{freq(s)}{N} \tag{7}$$

where $N$ is the total number of occurrences of all words in the corpus which are listed in the lexico-semantic hierarchy. From (6) and (7) it follows that synsets closer to the top of the hierarchy (i.e. that denote more abstract concepts) have higher probability but lower IC, i.e they are less informative. Synsets that are lower in the hierarchy (i.e. that denote more specific concepts) have higher IC.

In order to find how similar two synsets $s_1$ and $s_2$ are, we can look at the IC of $lca(s_1, s_2)$:

$$sim_{IC}(s_1, s_2) = -\log p(lca(s_1, s_2)) \tag{8}$$

If the $lca(s_1, s_2)$ has a high IC, it is low in the hierarchy and therefore the two synsets share many characteristics. In this paper, we use a variant of the similarity measure from (8), proposed in Lin (1998b):

$$sim_{lin}(s_1, s_2) = \frac{2 \times \log p(lca(s_1, s_2))}{\log p(s_1) + \log p(s_2)} \tag{9}$$

Here the IC of $lca(s_1, s_2)$ is normalised by the combined IC of the two synsets assuming they are independent. Thus, the Lin measure takes into account not only the common characteristics of the two synsets but also their differences.

CORNETTO provides probability values for each synset (see Figure 3). These values are obtained by counting the occurrences of all words in the hierarchy in a large Dutch corpus (approximately 500 million words). The provided probabilities make the calculation of the Lin measure straightforward.

Note that both evaluation metrics employed rely heavily on the *lca* of the two examined synsets: the higher the *lca*, the lower the similarity score for those synsets. The adjective synsets in CORNETTO are organised in a very flat hierarchical structure: The majority of these synsets are directly connected to the top node of CORNETTO. This makes the employment of the two evaluation metrics with respect to adjectives uninteresting, uninformative, and even unreliable. No matter how related two adjectives are, the application of both metrics is always guaranteed to result in low scores. That is why in our experiments adjectives are not taken into account.

### *4.3 Test set and corpora*

Since we focus only on nouns and verbs in our experiments, we select from the test set described at the end of Section 3.2 only those words which have been assigned noun and/or verb types by the LA method. This is the case for 374 words out of the 500 test words.[5] Next, 309 words out of the 374 words selected have been found in CORNETTO (204 nouns and 105 verbs). The test set for our experiments consists of these 309 words. It is important to note that these words are morphologically unambiguous. Further, the LA

[5] The remaining 126 words have been assigned adjective types only and therefore those words are not considered in our experiments.

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

method predicted the right type of POS for all 309 words. We also remind that for the purposes of our experiments, the words are temporarily removed from the Alpino lexicon.

Next, we randomly extract from CORNETTO 10,000 nouns and 2,500 verbs.[6] Out of these 12,500 words, the semantic space models we employ select the word which is most similar to a given test word.

The context features (context words (CW) and dependency relations) for the words taken from CORNETTO and the 309 test words are extracted from the Dutch Twente newspaper corpus and the Flemish Mediargus newspaper corpus (approximately 1.5 billion words). Then the co-occurrence matrices of words and features are built for each semantic model.

In order to obtain syntactic dependencies, we have parsed the corpora under two conditions. First, it is parsed with the standard Alpino configuration. A built-in unknown word guesser, based on the morphological form of the unknown word, employs various heuristics to guess the correct lexical type(s) for that word. In this case, the type assigned to each occurrence of the 309 test words is predicted by the guesser, which in turn affects the analyses of the sentences containing test words and the dependency relations extracted from those analyses. In order to evaluate the effect the employment of LA has on the syntax-based semantic models, in the second parsing setup, all sentences which contain words from the test set (approximately 440,000 sentences) are re-parsed using the lexical types assigned to them by our LA method. We should note that the parser is very robust and it always produces an analysis. If there is no parse covering the complete input sentence, the parser finds all parses for each substring and then selects the best sequence of non-overlapping parses.

## 5 Results

In this section we compare the performance of three semantic space models:

- **Alpino + LA** – A syntax-based model where the dependency relations for the test words are extracted from sentences parsed with the lexical types acquired via LA.
- **Alpino-standard** – A syntax-based model where the dependency relations for the test words are extracted from sentences parsed with the standard Alpino configuration.
- **Context Words** – A bag-of-words model which uses as features two words either side of the target word.

With respect to the two syntax-based models, we should note that the context selection function selects dependency triples only, i.e. longer dependency paths are not considered as features for the models. Further, all dependency triples which include a given target word are considered in the models. Finally, the set of basic elements consists of dependency triples, i.e. the triples are employed directly as features in the syntax-based models.

Further, for the bag-of-words model, experiments with window sizes ranging from 1 to 10 have been performed.[7] However, the selected window size of 2 yielded the best results.

[6] All words are also found in the Alpino lexicon, i.e. those words are not unknown.
[7] Note that sentence boundaries were not crossed.

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

Table 2. *POS frequencies for the word pairs produced by the three semantic space models. The frequencies are given in the format 'test_word_POS-top_similar_word_POS'*

| Model | Noun–Noun | Verb–Verb | Verb–Noun | Noun–Verb |
|---|---|---|---|---|
| Alpino + LA | 203 | 84 | 22 | 1 |
| Alpino-standard | 204 | 25 | 80 | – |
| CW | 167 | 59 | 46 | 37 |

In all models, we use a feature cutoff of 2, i.e. a particular word-feature combination has to appear at least twice in order to be included in the co-occurrence matrix. After the application of this cutoff the CW model contained 99,756 features and each of the two syntax-based models contained 11,386 features. Experiments with cut-off values ranging from 1 to 20 have also been performed but we found that each of the three models achieved its best performance with a cutoff value of 2.

Different settings of the parameters might lead to better results. However, the purpose of this paper is not to improve the various semantic models but to compare their performance when LA is applied. The semantic models are employed as a 'black box,' and therefore when constructing those models, we use standard parameter settings and similarity measures that are widely used in the field.

After building the co-occurrence matrices for all models, for each word in the test set, we calculated the cosine similarity between its context vector and the context vectors of all 12,500 words extracted from CORNETTO. Then for each test word, we take the CORNETTO word with the highest similarity score. Thus, for each model, we obtained a list with pairs of the type '*test word-most similar word*.'

Table 2 shows the number of POS combinations for the pairs found. Note that the total number of pairs for the Alpino + LA model sums up to 310, instead of 309. The model identified '*kussen*' as the most similar word to the test word '*zoenen*.' The meaning of both words is 'to kiss,' but '*kussen*' is also the plural form of the noun '*kus*' (pillow). That is why, it is mapped to both the verb and the noun entry for it in CORNETTO.

For each of the 204 nouns in the test set, the syntax-based models identify another noun as the most similar word (apart from one case for the Alpino + LA model). The crucial difference between the two models is in the number of verbs which have another verb as their most similar word. This is due to the fact that the Alpino unknown word guesser tends to assign noun types to most of the unknown words it encounters. Therefore, the features for the test words in the standard Alpino model are mostly syntactic dependencies between two nouns. This makes the standard model more biased towards identifying nouns as most similar words. Compared to the syntax-based models, the CW model shows a more balanced distribution. In addition, for the thirty-seven nouns from the test set, the model identifies a verb as the most similar word.

Before we perform the automatic evaluation of the obtained pairs against CORNETTO, we should make a decision how to treat those in which the two words have different POS. Intuitively, one would think that the word most similar to a noun is another noun. The same intuition is also valid for verbs. Indeed, the manual examination of all produced pairs of the type *test_word_noun-most_similar_word_verb* revealed that the involved words are completely unrelated. Therefore, one could argue that such pairs should simply be treated

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

Table 3. *Results for all test words for which scores can be calculated*

| Model | Wu & Palmer | | Lin | |
|---|---|---|---|---|
| | Number of words | Score | Number of words | Score |
| Alpino + LA | 240 | **.506** | 248 | **.395** |
| Alpino-standard | 250 | .425 | 255 | .324 |
| CW | 221 | .466 | 228 | .369 |

Table 4. *Detailed results for each POS combination*

| POS combination | Model | Wu & Palmer | | Lin | |
|---|---|---|---|---|---|
| | | Number of words | Score | Number of words | Score |
| Noun–Noun | Alpino + LA | 166 | **.507** | 168 | **.378** |
| | Alpino-standard | 168 | .488 | 170 | .355 |
| | CW | 140 | .491 | 142 | .366 |
| Verb–Verb | Alpino + LA | 58 | **.562** | 64 | **.475** |
| | Alpino-standard | 16 | .398 | 19 | .378 |
| | CW | 46 | .535 | 51 | .460 |
| Verb–Noun | Alpino + LA | 17 | **.289** | 17 | **.242** |
| | Alpino-standard | 66 | .272 | 66 | .230 |
| | CW | 35 | .268 | 35 | .236 |

as wrong. The same argument is valid for the verb–noun pairs. Following this reasoning, we can see that the CW model produced the largest number of wrong pairs, namely eighty-three. Though it delivered fewer wrong pairs, the standard Alpino model follows closely with eighty wrong cases. On the other hand, our model produces only twenty-three wrong pairs. The quality of the lexical entries acquired via LA allows for higher parsing accuracy, which, in turn, leads to a reduced number of wrong word pairs produced by the Alpino + LA model.

But then, during the manual examination of the verb–noun pairs, we identified a pair with a strong semantic relation between the words in it: *applaudisseren-handgeklap* (to applaud-hand clapping). This pair was produced by all three models. That shows that there could be cases where words with different POS can be semantically related and this relatedness is strong enough to show up in both syntax-based and bag-of-words models. That is why, we decided to include pairs containing words with different POS in the automatic evaluation as well.

We employ the Wu & Palmer and Lin measures to evaluate the similarity pairs we obtained against CORNETTO. Tables 3 and 4 show the results. The similarity score for each model is the average of the similarity scores for all pairs of words for which scores can be calculated. We should also note that, for polysemous words, we calculate the score based on the pair of synsets linked by the shortest path. This is valid for all results presented.

It is not possible to calculate the scores for all pairs. Calculation of the similarity measures is impossible for pairs where one of the words (or both) is not linked to any other synset in the hierarchy because in such cases there is no *lca*. There are also pairs for which an *lca* is found but it is not connected to the top of the hierarchy. For such pairs, the

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

Table 5. *Scores for pairs for which the Wu & Palmer and Lin measures can be calculated in all models*

| Model | Wu & Palmer | | Lin | |
|---|---|---|---|---|
| | Number of words | Score | Number of words | Score |
| Alpino + LA | | **.493*** | | **.393*** |
| Alpino-standard | 192 | .427 | 195 | .326 |
| CW | | .479 | | .361 |

Lin score can still be calculated but it is impossible to determine the Wu & Palmer score because it needs the path length between the *lca* and the top of the hierarchy. Naturally, such issues can be expected given that CORNETTO is being constructed semi-automatically from various lexical resources. The large number of missing links further indicates that low frequency words, as the ones in our test set, pose problems for CORNETTO as well.

Note that while scores for some verb–noun pairs could be calculated, this is not the case for all noun test words for which the most similar word is a verb. Most of the verb synsets in CORNETTO are connected to one of the following two synsets: 'passeren:verb:4' (to happen) and 'zijn:verb:7' (to be). In turn, the two synsets are connected to the top of the hierarchy. However, there are quite a few verbs the synsets of which are not connected to either of those two 'main' synsets. It so happens that in our experiments all verbs which are identified as being most similar to a given test noun belong to this group of 'unconnected' verbs and therefore similarity scores for noun–verb pairs cannot be calculated.

The results show that the Alpino + LA model outperforms the other two models. The difference in performance is most evident in the case of verb pairs as well as for pairs where the test word is a verb and its most similar word is a noun.[8] Again, the quality of the lexical types obtained for the test words via LA allows Alpino to produce more accurate parses, which, in turn, improves the quality of the extracted dependency relations for verbs.

While the improvement for verb and verb–noun pairs could be expected, the higher scores for noun pairs achieved by the Alpino + LA model compared to the CW model is a very positive observation. Usually, bag-of-words models are very robust and perform well at finding semantically similar nouns.

It is also interesting to investigate the nature of the relations discovered by the models. We found out that the bag-of-words model tends to identify tightly related words. For example, for *toekomstverwachting* (expectations for the future) the model suggests *prognose* (prognosis) as the most similar word. Another example is the word *fanaticus* (fanatic, zealot), for which *fanatiekeling* is suggested. The two words have virtually the same meaning. The Alpino + LA model, on the other hand, tends to discover topically related words. For example, it identifies *marktsituatie* (market situation, the current state of the market) as the most similar word for *toekomstverwachting*, and *ayatollah* (ayatollah) is suggested for *fanaticus*.

Another way of performing evaluation is to look only at pairs for which we are able to calculate Wu & Palmer and Lin scores for all three models. The average scores we obtain

[8] The similarity scores for verb–noun pairs are low due to the fact that the *lca* of a verb and a noun is the top of the hierarchy.

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

Table 6. *Detailed results for pairs for which the Wu & Palmer and Lin measures can be calculated in all models*

| POS combination | Model | Wu & Palmer | | Lin | |
|---|---|---|---|---|---|
| | | Number of words | Score | Number of words | Score |
| Noun–Noun | Alpino + LA | 128 | **.505*** | 129 | **.377*** |
| | Alpino-standard | | .494 | | .356 |
| | CW | | .498 | | .368 |
| Verb–Verb | Alpino + LA | 11 | **.552*** | 12 | .412 |
| | Alpino-standard | | .445 | | .383 |
| | CW | | .538 | | **.415** |
| Verb–Noun | Alpino + LA | 8 | **.274** | 8 | .238 |
| | Alpino-standard | | .269 | | .235 |
| | CW | | .273 | | **.239** |

for these pairs are given in Tables 5 and 6. In this evaluation scenario the pairs evaluated are the same for all models. Therefore, we can test for statistical significance using paired t-tests. Significantly better results (significance level $p \leq 0.05$) are marked with '*' in the tables.

Apart from the Lin score for verb pairs, our model achieves once more the best performance. The scores our model obtains for verb–noun pairs and the Lin score for verb pairs are virtually the same as the scores of the CW model. Although the differences between all other scores are smaller than the ones reported in Tables 3 and 4, our model still achieves the best performance.

The standard Alpino model has the worst performance in all scenarios. Thus, we can conclude that a syntax-based model is heavily affected by the lack of appropriate syntactic information about unknown words. In such cases, a bag-of-words model proves to be more robust since it does not consider any syntactic properties. However, if our LA method is applied, parsing accuracy increases and syntax-based models improve their performance. This makes the combination of these models with LA as best equipped to deal with the problem we investigate.

## 6 Conclusion

In this paper we compared the performance of two types of semantic space models – bag-of-words and syntax-based – when applied to the task of learning the semantics of unknown words. Their performance was examined in a case study involving a test set of words not listed in the lexicon of the Dutch Alpino grammar. We developed three semantic space models – one bag-of-words and two syntax-based models – which had the task of calculating the semantic similarity (in terms of cosine similarity) between each word in the test set and all words in a sample set extracted from the CORNETTO lexico-semantic hierarchy of Dutch. For each test word, the word with the highest similarity was selected from the sample set and the resulting word pairs were automatically evaluated against CORNETTO.

The bag-of-words model used as features two words on either side of the target word. The features for one of the syntax-based models were extracted from corpora parsed with

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

the default configuration of Alpino. The features were in the form of dependency triples. Then we applied our LA method (Cholakov and van Noord 2010) to learn lexical entries for the unknown words forming our test set. The sentences containing those words were re-parsed with Alpino and the newly acquired lexical entries. We extracted dependency triples which were employed as features in the second syntax-based model.

The bag-of-words model performed better than the model which involved the default Alpino configuration. The parser often produced wrong analyses for the sentences containing the unknown words which affected the quality of the features used in the syntax-based model. However, the syntax-based model which involved LA had the best performance. The high quality of the LA method allowed for this model to employ more accurate syntactic features.

We conclude that the combination of syntax-based semantic space models with an appropriate LA method is best suited for the task of learning the semantics of unknown words.

## References

Almuhareb, A., and Poesio, M. 2004. Attribute-based and value-based clustering: an evaluation. In *Proceedings of EMNLP 2004*, Edinburgh, UK, pp. 158–65.

Baldwin, T. 2005. General-purpose lexical acquisition: Procedures, questions and results. In *Proceedings of the Pacific Association for Computational Linguistics*, Tokyo, Japan, pp. 23–32.

Barg, P., and Walther, M. 1998. Processing unknown words in HPSG. In *Proceedings of the 36th Conference of the ACL*, Montreal, Quebec, Canada, pp. 91–5.

Berry, M. W., Dumais, S. T., and O'Brien, G. W. 1994. Using linear algebra for intelligent information retrieval. *SIAM Review* **37**: 573–95.

Cholakov, K., Kordoni, V., and Zhang, Y. 2008. Towards domain-independent deep linguistic processing: ensuring portability and re-usability of lexicalised grammars. In *Proceedings of COLING 2008 Workshop on Grammar Engineering Across Frameworks (GEAF08)*, Manchester, UK, pp. 57–64.

Cholakov, K., and van Noord, G. 2009. Combining finite state and corpus-based techniques for unknown word prediction. In *Proceedings of the 7th Conference on Recent Advances in Natural Language Processing (RANLP)*, Borovets, Bulgaria, pp. 60–65.

Cholakov, K., and van Noord, G. 2010. Acquisition of unknown word paradigms for large-scale grammars. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING-2010)*, Beijing, China, pp. 153–61.

Cholakov, K., van Noord, G., Kordoni, V., and Zhang, Y. 2011. Adaptability of lexical acquisition for large-scale grammars. In *Proceedings of the 8th Conference on Recent Advances in Natural Language Processing (RANLP)*, Hissar, Bulgaria, pp. 355–62.

Church, K. W., and Hanks, P. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics* **16**(1): 22–9.

Copestake, A., and Flickinger, D. 2000. An open-source grammar development environment and broad-coverage English grammar using HPSG. In *Proceedings of the 2nd International Conference on Language Resource and Evaluation (LREC 2000)*, Athens, Greece.

Crysmann, B. 2003. On the efficient implementation of German verb placement in HPSG. In *Proceedings of RANLP 2003*, Borovets, Bulgaria.

Curran, J. R., and Moens, M. 2002. Improvements in automatic thesaurus extraction. In *Proceedings of the ACL 2002 Workshop on Unsupervised Lexical Acquisition*, Philadelphia, PA, pp. 59–66.

Erbach, G. 1990. Syntactic processing of unknown words. IWBS Technical report 131, IBM, Stuttgart.

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

Erk, K. 2007. A simple, similarity-based model for selectional preferences. In *Proceedings of the 45th ACL Meeting*, Prague, Czech Republic, pp. 216–23.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: The MIT Press.

Fouvry, F. 2003. Lexicon acquisition with a large-coverage unification-based grammar. In *Companion to the 10th Conference of EACL*, Budapest, Hungary, pp. 87–90.

Golub, G. H., and Van Loan, C. F. 1996. *Matrix Computations*, vol. 3. St Baltimore, MD: Johns Hopkins Univ. Press.

Grefenstette, G. 1994. *Explorations in Automatic Thesaurus Discovery*. New York: Springer.

Horák, A., Vossen, P., and Rambousek, A. 2008. The development of a complex-structured lexicon based on WordNet. In *Proceedings of the 4th International Global WordNet Conference (GWC-2008)*, Szeged, Hungary, pp. 200–8.

Lin, D. 1998a. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th International Conference on Computational Linguistics*, Montreal, Canada, pp. 768–74.

Lin, D. 1998b. An information-theoretic definition of similarity. In *Proceedings of the 15th International Conference on Machine Learning*, Madison, WI, pp. 296–304.

Lowe, W. 2001. Towards a theory of semantic space. In *Proceedings of the 2nd Annual Conference of the Cognitive Science Society*, Edinburgh, UK, pp. 576–81.

Malouf, R. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the 6th conference on Natural Language Learning (CoNLL-2002)*, Taipei, Taiwan, pp. 49–55.

McCarthy, D., Koeling, R., Weeds, J., and Carroll, J. 2004. Finding predominant word senses in untagged text. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, pp. 279–86.

Miller, G.A., and Charles, W. G. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes* **6**(1): 1–28.

Ordelman, R. J. F. 2002. Twente nieuws corpus (TwNC). Technical report, Parlevink Language Technology Group, University of Twente, Enschede, Netherlands.

Padó, S., and Lapata, M. 2007. Dependency-based construction of semantic space models. *Computational Linguistics* **33**(2): 161–99.

Rapp, R. 2004. A freely available automatically generated thesaurus of related words. In *Proceedings of the 4th Language Resources and Evaluation Conference (LREC 2004)*, Lisbon, Portugal, pp. 395–8.

Rothenhäusler, K., and Schütze, H. 2009. Unsupervised classification with dependency-based word spaces. In *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, Singapore, pp. 17–24.

Salton, G., Wong, A., and Yang, C. S. 1975. A vector space model for automatic indexing. *Communications of the ACM* **18**: 613–20.

Schütze, H. 1998. Automatic word sense discrimination. *Computational Linguistics* **24**(1): 97–123.

Turney, Peter D., and Pantel, P. 2010. From frequency to meaning. Vector space models of semantics. *Journal of Artificial Intelligence Research* **37**(1): 141–88.

Van de Cruys, T. 2008. A comparison of bag of words and syntax-based approaches for word categorization. In *Proceedings of the ESSLLI Workshop on Distributional Lexical Semantics. Bridging the Gap Between Semantic Theory and Computational Simulations*, Hamburg, Germany, pp. 47–54.

Van der Plas, L., and Tiedemann, J. 2006. Finding synonyms using automatic word alignment and measures of distributional similarity. In *Proceedings of the COLING-ACL Joint Conference*, Sydney, Australia, pp. 866–73.

van Noord, G. 2006. At last parsing is now operational. In *Proceedings of TALN*, Leuven, Belgium, pp. 20–42.

Vossen, P. (ed.) 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Dordrecht, Netherlands: Kluwer.

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press

Wu, Z., and Palmer, M. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, Las Cruces, New Mexico, pp. 133–8.

Zhang, Y., and Kordoni, V. 2006. Automated deep lexical acquisition for robust open text processing. In *Proceedings of the 5th International Conference on Language Recourses and Evaluation (LREC 2006)*, Genoa, Italy, pp. 275–80.

https://doi.org/10.1017/S1351324913000053 Published online by Cambridge University Press