UNIVERSITÉ DE LORRAINE | UFR MATHÉMATIQUES ET INFORMATIQUE

rijksuniversiteit groningen / faculteit der letteren

European master in Language and Communication Technologies

Université de Lorraine & Rijksuniversiteit Groningen

Resolving PP-attachment ambiguity in French with distributional methods

MSc thesis, academic year 2011/2012

Author: Simon Šuster

Supervisors: prof. dr. Christophe Cerisara, main supervisor
prof. dr. ir. John Nerbonne, co-supervisor

June, 2012

Acknowledgments

I would like to thank my supervisors for discussions, especially I would like to thank Christophe for his outstanding guidance, constant availability, the attention with which he listened to my ideas and the help he provided when I found myself in difficulties. Moreover, I am also grateful for having been provided an excellent working place at the Loria laboratory.

Next, I would like to thank the Slovene Human Resources Development and Scholarship Fund for supporting financially my studies during the past two years.

Finally, I wish to express my thanks to my family: Olga, for supporting me morally, and generously providing me with the time needed to complete both the studies and the thesis; my parents and my grandmother, who also contributed to a more pleasant stay abroad.

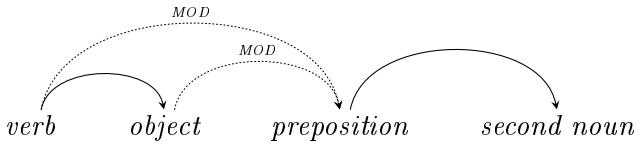
Contents

Chapter 1

Introduction

> Clearly much of the insight into word-meaning is to be gained by observing the ways in which words are strung together by competent practitioners of a language. However, the approach has its limitations. Semantics done this way has more of the character of an "observational science", like geology, than that of an "experimental science", such as physics or chemistry. Scientists of the former type are, in a sense, at the mercy of the phenomena they study; what happens, and when, and under what conditions is largely beyond their control.
>
> [Cruse 1986]

A persistent problem in natural language parsing is resolving attachment ambiguities. In building a syntactic tree of a sentence, the problem is in deciding the correct attachment of a phrase between at least two possible attachment sites. Perhaps the most ubiquitous and researched problem in this field is the prepositional-phrase ambiguity (PPA, in the rest of the thesis) resolution. The following example is its illustration in dependency parsing:



Example 1.

In its basic form, the task is to decide whether the PP, formed of a preposition and a noun, should modify the noun or the verb (dotted arcs in the example 1).

The majority of approaches to resolving the PPA ambiguity use some form of lexical information. From the example above, it is clear that the problem can be largely tackled partly by accounting for the identity of the preposition and partly by accounting for the identity of the noun following it. During the past 20 years or so, the research has chiefly focused on solving the problem in isolation, i.e. given a set of ambiguous cases, decide whether the correct attachment ought to be verbal

or nominal. Recently, this approach has received criticism, and a new direction for research was proposed which would seek solutions based on the parser output or integrating in the parsing procedure. In our work, we make an effort to study PPA disambiguation in this "natural" setting. This is achieved by proposing a method that works on the output of the parser and also integrates with the parser by predetermining the dependency links for the hard cases in the text that is subsequently analyzed by the parser. Our method would ideally correct (some of) the decisions made by the parser, hopefully resulting in improved overall parsing accuracy. If we do not succeed in actually improving the parsing results (which is possible due to likely low impact), this could lead us towards improved understanding of how distributional(-semantic) information can be used towards this end and in structural disambiguation in general.

To achieve our goal, we use lexical information gathered from large corpora, and without the need for human annotation of PP-attachment cases. This makes our approach unsupervised[1] and completely data-driven. We develop two distinct models capable of resolving attachment ambiguities. The first model makes use of the selectional preferences between the preposition and the competing attachment sites (verb and first noun). In the second case, we create a a vector-space representation of the contexts/meaning of the words, which allows to introduce the identity of the second noun into our model.

Despite the proliferation of the work on the PP-attachment disambiguation, our original contribution is that we propose and evaluate a completely distributional way of complementing the parser (parser's output) with the type of lexical information that is usually not captured/exploited in (statistical dependency) syntactic parsing, and with, for one model, accepting the distributional hypothesis and taking the step towards (lexical) semantics. We also view our task as detection (due to the attachment data distribution) rather than the perspective traditionally taken in the PPA disambiguation research – the classification. We believe that it is only fair to focus on the type of PPA cases that are really problematic for the parser, thus discarding prepositions that are straightforward for the parser ("de" in French). Since this preposition is the most frequent in the language, the reported accuracy figures in the past research in reality reported mostly the accuracy for exactly this prepositions ("of" in English), which we see as a methodological drawback.

Understandably, a vast majority of the research has been performed on English. For other languages, such as German, Spanish, Dutch and French, some work exists, but is scarce. With taking French as our target language, we hope to contribute to understanding of the PPA disambiguation and its role in syntactic parsing specifically for French.

In the following section, we start by setting the context for our work: we briefly

[1] We note that the definitions of the term "unsupervised" vary in the literature. Here, we use the term to mean that we do not know the classification of the data in the training sample (contrary to supervised learning, where the status or classification of a piece of training data is known, for example because of human annotation). This definition is in line with [Manning & Schütze 1999, p. 232].

sketch the notions of dependencies, the types of dependency parsers and how we can evaluate their output. We then present the distributional semantic framework and techniques, together with the distributional hypothesis. In the second part of the next chapter, we describe in detail the problem of PPA, we summarize the approaches to disambiguation and finally provide an exhaustive survey of the literature on the subject. The second half of this thesis report is a detailed report of our experiments. Firstly, a quantitative analysis of the PP-attachment cases is performed on the gold annotated corpus, and an estimate of the parser errors is determined on the already parsed texts. Secondly, we present the methodology in our main experiments and evaluate the results. In the end, we summarize our findings and contributions in the conclusion.

Chapter 2

Background

2.1 Natural language processing

In this section, we set the theoretical background and introduce the subjects and techniques that relate to, or are used in, our approach to disambiguation. The goal is to provide an overview without going into unnecessary details that are not explicitly mentioned or dealt with in the rest of the thesis.

2.1.1 Dependency parsing

2.1.1.1 Foundations

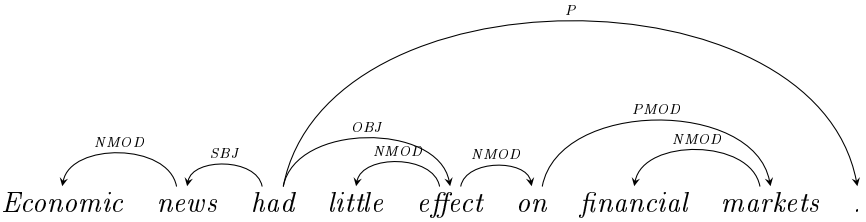
Syntactic parsing is an analysis of the structure of human language sentences. It is a key task in NLP, because it is a crucial step on the way to semantic processing [Jurafsky & Martin 2008]. Since natural language sentences are often ambiguous as to their syntactic structure, parsing is a hard task. It can be seen as a twofold task: *strict parsing*, which will determine the set of possible syntactic representations of a particular sentence, and a *disambiguation procedure* selecting among the set of alternatives the preferred candidate. With statistical (probabilistic) modeling, it is possible to determine the most plausible parse for a sentence in case of ambiguity. We call the parsing system that uses such knowledge a *statistical parser*.

On the level of the syntactic representations they encode, the syntactic parsing can be divided into two categories: *dependency parsing* vs. *phrase structure (or constituency) parsing*. Dependency parsers construct the dependency structure representing governor-dependent relations between words, usually complemented by functional classes such as subject and modifier. In phrase structure parsing, on the other hand, the representation groups words into phrases categorized with classes such as noun phrase or verb phrase. It is important to realize that despite this distinction, implicitly both categories can convey equivalent information as the conversion from one into the other is possible in most cases. Because it is the first type of parsing that is of primary interest in our case, we are leaving the topic of phrase structure parsing aside.

Dependency parsing is capable of achieving highly accurate results in the analysis of many languages and for many NLP tasks and applications [Kübler *et al.* 2009]. Two common observations by the proponents of this branch of parsing is that the predicate-argument representation is very intuitive and that it is especially suitable for languages with less fixed word order. It makes use of dependency grammar linguistic framework, in which words are linked by binary, asymmetrical relations

called *dependency relations* [Tesnière 1959]. A dependency relation's arguments are the *head* (or the *governor*) and the *dependent*. A dependent stands in a relation with the head if the head modifies it, and – in graphical representation – there is an arc pointing from the head towards the dependent. The following is an example of a dependency graph from the Penn Treebank (without POS-tags, taken from [Nivre 2005]):

Example 2.



There exists another division of parsing systems depending on the provenance of the (possible) grammatical rules, namely that of *grammar-driven* and *data-driven parsing*. An approach is grammar-driven if a formal grammar was crafted and is used in the construction of possible analyses of the sentence.[1] Parsing is data-driven when the model (or grammar) is induced by means of machine learning techniques from the previously annotated corpus data (a *tree-bank*). The two types are not exclusive, however, since there exist parsers combining both, two examples being the probabilistic context-free grammar parser of Collins [Collins 1997] and the Dutch Alpino parser which joins an attribute-value grammar with maximum entropy disambiguation. An example of a completely data-driven parsing system is the MATE parser [Bohnet 2010], which we use in our present work and present in more detail in the next section. As is usual with data-driven approaches in NLP, they can vary as to the amount of supervision included in the learning. Here, we only deal with supervised dependency parsing, although applications of structural disambiguation with distributional (semantic) methods are well-worth exploring also in unsupervised parsing. The supervised dependency parsing consists of two different problems, namely that of *learning*, which is about induction of a parsing model given a training set of sentences, and *parsing*, where the goal is to arrive at an optimal dependency graph given the model and a sentence [Kübler *et al.* 2009]. The model type, and the algorithms for learning the model and for parsing sentences let us discriminate between two types of data-driven dependency parsers, *transition-based* and *graph-based*. Perhaps the best-known representatives of the two types are the Malt parser [Nivre 2006] and the MSTParser [McDonald *et al.* 2005]. These are the parsers regularly achieving among the best results in parser comparison events such as the CONLL 2007 Shared Task on Dependency Parsing [Nivre *et al.* 2007].

[1] Note that the formal grammar, when it is crafted by a linguist, can be data-driven or not.

2.1.1.2 Parsing systems and parser evaluation

This section introduces the parsing system we use in our research and the French language corpora used for training and testing both the parser and our PPA disambiguation methods.

We use the MATE parser [Bohnet 2010] which is part of the MATE tools available at `http://code.google.com/p/mate-tools/`. The parser is integrated in the JSafran software [Cerisara & Gardent 2009] which allows easy and fast training (and several other tasks including annotation) of a model. In our case, the parsing employs POS-tags obtained by the TreeTagger [Schmid 1994]. The MATE parser is a state-of-the-art graph-based dependency parser that uses as its base the maximum spanning tree dependency parsing algorithm in combination with the passive-aggressive perceptron algorithm. It achieved LAS of 90.33 for English and 88.13 for Spanish, for example, on the CONLL 2007 data-set, thus beating other competing systems. Due to a parallelization algorithm, the very good time efficiency makes it appropriate for running relatively quick experiments.

Dependency parsers are normally tested by parsing a part of the tree-bank and comparing the parses to the gold standard annotations [Kübler *et al.* 2009]. The following evaluation metrics are usually reported:

- Attachment score: the percentage of words with correct heads. Another variant takes the average of the percentage of words with correct heads for all the sentences.
- Precision: the percentage of dependencies of a certain type *in the parser output* that were correct
- Recall: the percentage of dependencies of a certain type *in the test corpus* that were correctly parsed
- F-measure: the harmonic mean of precision and recall

The above metrics can be:

- Labeled: considering heads and labels
- Unlabeled: considering heads only

The most commonly reported figures are the labeled attachment score (LAS) and unlabeled attachment score (UAS).

2.1.2 Distributional semantic modeling

In a broad sense, a word-space model, or a distributional semantic model (*DSM*, in the following), is a computational model of meaning of linguistic units that utilizes the distributional patterns collected over large corpus data in order to represent semantic similarity between these units in spatial proximity [Sahlgren 2006, Turney & Pantel 2010]. Most frequently, the DSMs deal with the linguistic units

on the word level, at least this level is the most researched, although it is becoming more and more common to encounter DSMs combining two or more words (*composition*), moving from words to the phrase and the sentence level [Mitchell & Lapata 2008, Mitchell & Lapata 2010, Baroni & Zamparelli 2010]. According to [Schütze 1993], it is by vector similarity that we have that vectors which are close represen semantically related words, and the other way around, vectors or words which are far apart are unrelated. This idea was aptly named "the geometric metaphor of meaning" by Sahlgren.[2] There are many ways of building a DSM, which is described in section 2.1.2.2. However, what all have in common is a type of distributional hypothesis, introduced in the next section.

DSMs are normally implemented as matrices. We reserve the term DSM only for matrices that include elements whose values are corpus (event) frequencies or some other values derived from them, thus reflecting distributional properties of words.

2.1.2.1 Distributional hypothesis

> "/.../ if we consider words or morphemes A and B to be more different in meaning than A and C, then we will often find that the distributions of A and B are more different than the distributions of A and C. In other words, difference of meaning correlates with difference of distribution." [Harris 1954]

> "You shall know the word by the company it keeps." [Firth 1957]

> "/.../ the meaning of a word is fully reflected in its contextual relations; in fact, we can go further, and say that /.../ the meaning of a word is constituted by its contextual relations." [Cruse 1986]

In this work, we accept the core idea, reflected in the above quotes, which underlies the word-space modeling and enables discussing *semantic* similarity, i.e. the *distributional hypothesis*:

> words with similar distributional properties have similar meanings

Different flavors of DSM will define "having similar distributional properties" in slightly different manner: this can mean, for instance, "occurring in similar/same documents" or "occurring in similar context windows of a particular size", depending on the parameter configuration in the DSM design. The assumption stated above is common not just in NLP applications, but other domains such as cognitive science, corpus linguistics and lexicography [Lenci 2008, Landauer *et al.* 2007, Kilgarriff 1997]. In a way also a function of the discipline in which the hypothesis is assumed, [Lenci 2008] distinguishes between the *weak* and the *strong* distributional hypotheses. The weak version is a quantitative method for analyzing semantic properties, emphasizing a *correlation* between semantic content and linguistic

[2] "Meanings are locations in a semantic space, and semantic similarity is proximity between the locations."[Sahlgren 2006, p. 19]

distributions: the distributional method can thus help us study different lexical semantic phenomena. In other words, it does not postulate that the meaning (as elusive as the term itself is) can be entirely captured by observing linguistic contexts. This version can be frequently seen in applications like word sense disambiguation, thesaurus construction and question answering. The strong version raises the first hypothesis up to the *cognitive* level: word distributions play a constitutive role in semantic representations. There is a causal relationship between being exposed to contexts of a particular word and abstract contextual representations for this word. This assumption has been used to model linguistic-psychological research, e.g. similarity judgments, semantic priming and child lexical acquisition. One particularly well-known materialization of a DSM which was originally designed with this version of distributional hypothesis in mind, is the Latent Semantic Analysis (LSA), arguably a cognitively plausible model for semantic representations. Its usability attracted the information retrieval community to adopt it as a particularly often used technique in retrieving documents.

We shall hereby accept the first, weak version of the distributional hypothesis: proximity in a word space model reflects semantic similarity between words or larger linguistic units. What interests us is the application of a DSM in a structural disambiguation task, in a very concrete task of PPA disambiguation, which we view as a sub-task of parsing. Since we do not pretend to carry out experiments with consequences or insights for the cognitive level, there is no reason nor need for us to adopt the stronger hypothesis. As it will become clear in the following chapters, we will claim that distributional properties and distributional semantic similarity between elements in an ambiguous PPA case could indicate what is the correct attachment type. But what exactly do we have in mind when we say "semantic similarity"? To answer this question, we note that this over-generalization of the term "similarity" is perhaps the most acute problem and the most criticized aspect of distributional semantics, which is undoubtedly related to the difficulty of the field to address essential issues in semantic representation such as compositionality, inference and reference [Sahlgren 2006, Padó & Lapata 2003, Lenci 2008]. Presently, however, this challenges are being more and more often addressed and tackled (see for example the First Joint Conference on Lexical and Computational Semantics and the SemEval task in 2012; see also the pioneering work on the subject of compositionality in distributional semantics by [Erk & Padó 2008, Mitchell & Lapata 2008]). We recognize that, with very simple DSMs, where contexts are represented as bag of words, we capture a very broad scope of relations that fall into the category of semantic similarity (synonymy, antonymy, hyponymy, meronymy etc.). This is usually criticized by taking the prescriptivist stance and approach the DSMs with a a priori division of notions of semantic similarity (see [Sahlgren 2006] for a more in-depth discussion). However, from a descriptive point of view, already a notion of broad semantic similarity seems perfectly acceptable, and it is also true that different (more sophisticated) implementations of DSM can readily outline specific semantic similarity types or relations.

2.1.2.2 Construction and parameters of DSMs

DSMs are models that can be conveniently represented with matrices. In this work, we use upper-case letters to denote matrices and lower case letters to denote vectors. A matrix A has $m * n$ dimensions. m stands for the number of "rows" (e.g. unique words or terms) and n for the total number of "columns" or "dimensions".

$$A_{m,n} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

Figure 2.1: Representation of a matrix.

The element $a_{1,2}$ of A thus represents a value for the row vector a_1, at the dimension 2.

Suppose we are interested in finding out similarities of some nouns occurring in a corpus (the example matrix is taken from [Evert & Lenci 2009]). We build our matrix representation of nouns occurring in a context window of some size by scanning through the corpus for nouns. When we find a noun, we increment the frequency count for each context word we encounter in the window. For example, we see the noun "dog", look at the words in the window, and increase the frequency by 1 if we find one occurrence of "bark". Eventually, we could end up with the following matrix

$$A_{|\text{w}|,|\text{context w}|} = \begin{array}{c} \\ dog \\ cat \\ lion \\ light \\ bark \\ car \end{array} \begin{array}{c} \begin{array}{cccccc} leash & walk & run & owner & leg & bark \end{array} \\ \begin{bmatrix} 3 & 5 & 1 & 5 & 4 & 2 \\ 0 & 3 & 3 & 1 & 5 & 0 \\ 0 & 3 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 & 1 & 0 \\ 0 & 0 & 4 & 3 & 0 & 0 \end{bmatrix} \end{array}$$

Figure 2.2: Example matrix A.

where row vectors are the nouns in the corpus and dimensions are context words for these nouns. We see that, for the noun "light", we did not find any occurrence with any context word, possibly because the noun does not occur in the corpus at all. The semantic content of a noun is thus represented as a row vector in A. Two nouns are similar if their vector representations are similar. There exist many methods for comparing vectors. One with which we can obtain a intuitive graphical representation of vector similarity is to first reduce the number of dimensions to 2 (for example by means of a dimensionality reduction such as Singular value decomposition), and then plot the vectors in a Cartesian plane. Since we only have two values for each vector, we can interpret them as coordinates x, y.

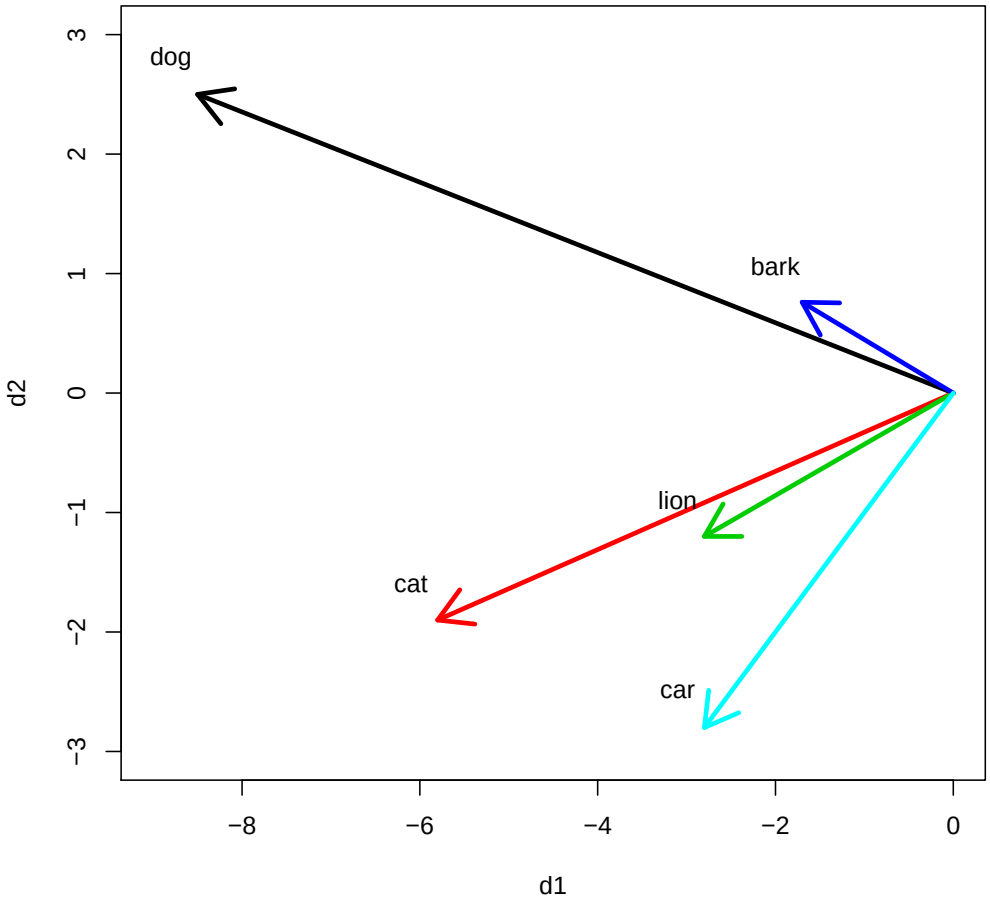


Figure 2.3: Vectors of the matrix A in a 2-dimensional plane after the reduction of A with the SVD

Note that the length of the vectors on the plot 2.3 is different and is a function of the frequency counts in the vector. Because of the application of the SVD, some values are negative, but this is not important for the interpretation here. However, what is of real interest is the direction of the vectors. Vectors pointing in a similar direction share a similar meaning, and vectors which are further apart are less semantically similar. "Bark" and "dog", as well as "cat" and "lion" are close to each other, while "car" points pretty much to its own direction, indicating semantic isolation. It is also almost orthogonal to "bark", suggestive of the lack of semantic similarity between two words.

In the matrix A, our dimensions were words obtained from context windows, while rows were single words. According to [Turney & Pantel 2010], one of the

fundamental distinctions in DSMs is the decision what will be represented by rows and columns. We can distinguish between (at least) three types of matrices:[3] *word-context*, *term-document* and *pair-pattern*. A word-context matrix is the one we have just presented as an example in this section. The main idea is that we look at the distribution of a word in a certain type of context (e.g., a window of a certain number of words to the left and the right, word's dependent or head in some specific syntactic relation ...). In a term-document matrix, a document vector represents the corresponding document as a bag of words. The number of vectors is the number of documents in a collection. In information-retrieval terms, the frequency of a word in a document would then indicate the relevance of the word/query to this document, and looking at the vector for a document would express what the document is about. In a pair-pattern matrix, rows correspond to word pairs, such as "policeman-gun" and "teacher-book", whereas columns are patterns in which pairs occur ("X uses Y" and "X with Y").

DSMs can be conveniently thought of as a tuple, which is a combination of a matrix with specific parameters [Evert & Lenci 2009]:

$$< T, C, R, W, M, d, S >$$

- T: target elements for which the DSM provides a contextual representation, i.e. rows
- C: contexts in which T occur, i.e. dimensions
- R: relation between T and C
- W: weighting scheme for the values of the matrix
- M: DSM matrix, $T \times C$
- d: dimensionality reduction function, $d : M \rightarrow M'$
- S: distance measure between vectors in M'

For each step in the construction of a DSM, various parameters should be decided upon. The C will influence the type and the extent of corpus processing needed in obtaining M. For instance, having a dependency parsed corpus, one can decide to only choose T and C in a particular syntagmatic relation R, say subject (as in [Padó & Lapata 2007]). The number of standard pre-processing procedures must be decided upon, too, such as tokenization, lemmatization (normalization) and POS-tagging. W and d get us to more mathematical steps. W deals with smoothing or weighting the raw frequency counts that would otherwise – following the Zipfian distribution[4] – provide us with a small number of very frequently occurring types and an immense number of very infrequently occurring types [Evert 2005]. Various

[3] Other representations than matrices are possible, for example higher-order tensors [Van de Cruys 2009].

[4] The frequency of the r-th most frequent type is proportional to 1/r.

weighting techniques have been proposed in the literature. The logarithm of the frequency

$$log_2(1 + M_{i,j}) \tag{2.1}$$

can be applied to each element of the matrix M, resulting in a dampening of high count events. In information retrieval, it is common to use the tf-idf weighting schemes [Manning & Schütze 1999] (which we do not define here, since we do not use them in our research). Another option is to use a version of the Poisson distribution to weight the value of the matrix. Finally, association measures can be used (we use them in our research) in order to put more emphasis to contexts which are significantly associated with a target word. Different association measures behave differently. Pointwise mutual information (PMI) is a score obtained after taking the logarithm of the ratio between the observed co-occurrence probability of the word with the context and the expected co-occurrence of the two [Church & Hanks 1990]:

$$PMI(w_1, w_2) = log_2 \frac{P(w_1, w_2)}{P(w_1)P(w_2)} \text{ where } P(w_1, w_2) = \frac{|w_1, w_2|}{N} \text{ and } P(w) = \frac{|w|}{N} \tag{2.2}$$

The score can take any value, but is zero when w_1 and w_2 are independent (not sharing any information). Because of its lack of a fixed upper bound, it is not possible to say when words are perfectly correlated. It thus allows only for relative comparisons.

One well-known drawback of PMI, which is in our view frequently neglected in the literature, is that it overestimates the importance of very rare words. To realize the sensitivity of PMI to data-sparseness, we consider the case when w_1 and w_2 are maximally associated. Two words are maximally associated when they only occur with each other (for each occurrence of w_1, we know that it co-occurs with w_2 and that we have found a w_1w_2 pair). The probability of encountering either w_1, w_2 or w_1w_2 are thus the same:

$$P(w_1, w_2) = P(w_1) = P(w_2) \tag{2.3}$$

and the above PMI formula becomes:

$$log_2 \frac{P}{P^2} = log_2 \frac{1}{P} \tag{2.4}$$

In other words, the lower the probability of a word, the higher the PMI. Several variants of PMI have been proposed to alleviate this problem. One is to multiply the PMI with the observed probability [Evert 2005]:

$$Local\text{-}PMI(w_1, w_2) = P(w_1, w_2) log_2 \frac{P(w_1, w_2)}{P(w_1)P(w_2)} \tag{2.5}$$

which was found to perform well in practice [Baroni & Zamparelli 2010] and it is actually a correlate (a sub-term) of the Log-Likelihood Ratio [Dunning 1993]. Com-

pared to the plain PMI, the formula will additionally weight the PMI score depending on how strong is the observed evidence for the co-occurrence. In other words, very infrequent word combination receive less importance than frequent ones.

In addition to the PMI and LPMI, we use another variant of the PMI in the experimental part, namely the Positive PMI (PPMI). It is calculated identically to the PMI, except that the negative values, which are produced with the plain PMI formula as a result of taking the logarithm (when the fraction part is smaller than 1), are mapped to 0. Negative values indicate the lack of association between the two words. The lower bound of the PPMI thus becomes 0, which makes it more handy for operations on matrices and vector comparison. [Bullinaria & Levy 2007] found out that the PPMI outperformed several other weighting schemes in a task of measuring semantic similarity with a word-context DSM.

A wide choice of metrics is available (step S) also for measuring similarity or distance between vectors: the dot (also scalar) product, the Euclidean distance,[5], the cosine of the angles between two vectors, Jenson-Shannon divergence and relative entropy are some of them. The cosine between two vectors is probably the most often employed metric in vector-space modeling. It is also the one we use throughout our experiments. The calculation takes the scalar product of the vectors and then divides it by their norms:

$$sim_{COS}(\vec{x}, \vec{y}) = \frac{x \cdot y}{|x||y|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \tag{2.6}$$

In addition to the relative efficiency of calculating the Cosine metric, its attractiveness lies in the fact that it is easily interpretable, since it ranges from 1 for maximally similar vectors (actually perfectly aligned) over 0 for dissimilar, orthogonal (right-angle) vectors, to –1 for vectors pointing in opposite directions [Widdows 2004, Sahlgren 2006]. Note that when vectors include only positive values, the lower bound is 0. From our example vectors from the matrix A (2.2), the Cosine between "cat" and "lion" is 0.8, while the Cosine between "cat" and "car" is 0.45.

With respect to the step d, the dimensionality reduction is often opted for when constructing large DSM. There are two problems encountered when manipulating very large matrices, one is the number of dimensions that can be in order of millions for present-day corpora and the number of matrix elements with value 0, the so-called sparsity problem. Since a statistical model requires a lot of data as evidence, having a bigger DSM is generally a highly valued asset. However, this comes at the expense of scalability and efficiency of any operation carried out on the matrix.[6] The sparsity problem relates to the fact that, in a matrix without reduced number of

[5] Both are sensitive to the very frequent words, so the effects of vector length should be eliminated before applying them

[6] Consider a matrix built from lemma types in the British National Corpus with frequency above 10 [Evert & Lenci 2009]. This yields a 83,926 by 83,926 matrix (7 billion elements), and with 8-byte float data type for matrix elements, we would need 56GB of RAM to load it in the memory. 99,68% of such a matrix would be populated with zeros.

dimensions, the total number of 0-valued elements outnumbers the non-zero values, which is a consequence of the Zipf law (most of the words combine with a small set of words, i.e. behave non-promiscuously). A partial solution to data sparseness is to use a special compact sparse matrix format for storing the matrix. In such a representation, only non-zero elements are stored linked to their respective indices. A simple solution within dimensionality reduction is to select only specific dimensions based on the frequency criterion. Looking at the frequency distribution for context words, either the head or the tail, or both, can be pruned with a certain threshold. For example, taking N number of most frequent words would remove "noise" from low-frequency events, while removing the Q number of most frequent words would eliminate the "function" words, i.e. words with little semantic discriminative power. This could drastically reduce the size of the matrix and speed up computations. This technique is often called the feature selection.[7] A similar operation can be performed on matrix rows, where only the words/terms of interest to the research are considered.

In another approach, the number of dimensions is reduced as well, except that the vector representations for all dimensions are changed (vectors are mapped to a sub-space). These techniques include Singular value decomposition (SVD), Principal component analysis (PCA) and Random Indexing (RI), among others. We hereby introduce SVD, since it is the only dimensionality reduction technique used in our experiments.

SVD is a technique rooted in linear algebra for deriving the matrix $\hat{A}$ from the product of the factorization of the original matrix A. SVD can be seen as data smoothing, extreme values being mapped to less extreme, and at the same time solving the sparsity of data problem by creating a dense matrix. To illustrate the technique, A is decomposed into three linearly independent[8] matrices:

$$\underset{m}{\overset{n}{\left[\begin{array}{c} A \end{array}\right]}} = \underset{m}{\overset{m}{\left[\begin{array}{c} U \end{array}\right]}} \cdot \underset{m}{\overset{n}{\left[\begin{array}{c} \Sigma \end{array}\right]}} \cdot \underset{n}{\overset{n}{\left[\begin{array}{c} V^T \end{array}\right]}}$$

Figure 2.4: SVD decomposition.

U is the matrix of left singular vectors with the same number of rows as A, but with m number of dimensions. V is the matrix of right singular vectors with the same number of columns as A, but with n number of rows. Σ is the matrix of singular values on the diagonal ($\sigma_1 \ldots \sigma_m$) with other elements being zero. All σs are ordered from the highest value to the lowest, essentially representing the amount

[7] We are aware of the fact that with feature selection the information from discarded dimensions is completely lost. But we assume that the information was not relevant or was negligible in the first place.

[8] They are orthogonal to each other. Simply put, this means that no matrix can be derived from the other two.

of variance captured by these latent dimensions (how significantly they contribute to the product). If we omit some singular values when reconstructing the original matrix, the $\hat{A}$ will be the best possible approximation of A in the lower dimensional space. Reducing Σ to k highest singular values and adapting the dimensions of U and V correspondingly is commonly called *truncated* SVD (see [Deerwester *et al.* 1990] for an application on a word-context DSM):

$$\underset{m}{} \overset{k}{\begin{bmatrix} \\ U \\ \\ \end{bmatrix}} \cdot \underset{k}{} \overset{k}{\begin{bmatrix} \\ \Sigma \\ \\ \end{bmatrix}} \cdot \underset{k}{} \overset{n}{\begin{bmatrix} \\ V^T \\ \\ \end{bmatrix}} = \underset{m}{} \overset{n}{\begin{bmatrix} \\ \hat{A} \\ \\ \end{bmatrix}}$$

Figure 2.5: Truncated SVD.

Multiplying only the truncated U and Σ will produce a matrix with only k dimensions. Finally, it is often argued that applying a dimensionality reduction such as SVD can improve the quality of a semantic space [Landauer *et al.* 2007]

To sum up, in this section we listed and introduced some of the possible parameters in DSM construction, but by no means all, and the interested reader should consult [Turney & Pantel 2010] or [Evert & Lenci 2009] for a more complete picture. It is the combination of parameters that determines the DSM flavor, each known under a different name. Here are some examples: Latent Semantic Analysis (LSA) for applying the SVD to log-frequency weighted term-document matrices in information-retrieval and cognitive science research [Landauer *et al.* 2007]; Hyperspace Analogue to Language for a plain-frequency word-context DSM which implements a context window that distinguishes between left and right contexts [Lund & Burgess 1996]; Dependency Vectors for dependency-based contexts in a word-context DSM with the log-likelihood ratio for weighting [Padó & Lapata 2007]; and Distributional Memory for dependency-based contexts with LPMI weighting.

2.1.2.3 Selected advantages of distributional semantic models

Lastly, we would like to answer the question why using DSM is an attractive choice compared to using already existing human-built inventories. Firstly, they require less labor in construction than the human-built ontologies such as WordNet. This way, they can be more readily used for languages with less developed language technology infrastructure, or for different domains, provided of course that a text corpus can be created. Secondly, they have an increased coverage over human-built resources; they are less affected by the data sparseness. Thirdly, they are completely data-driven, while this is not necessarily so for human-built knowledge bases that are sometimes affected by the bias of introspective and prescriptivist judgments. Fourthly, they are not limited to any set of grammatical category, and easily provide information about, say, prepositions and conjunctions, which are usually omitted in the inventories such as WordNet. Finally, they perform well on a variety of tasks in

NLP [Turney & Pantel 2010].

2.2 Structural disambiguation: the case of PP-attachment

"Your dog's been chasing a man on a bicycle."
"Don't be stupid! My dog can't ride a bike."

Structural disambiguation is necessary whenever a sentence can be parsed in two or more different ways. Attachment ambiguities are the most common structural ambiguities, and the prepositional-phrase attachment is one of the most common attachment ambiguities in many languages, it is definitely best studied in English. We hereby concentrate only on the problem of PPA ambiguity and leave other types of structural and attachment ambiguities aside.

2.2.1 The problem of prepositional-phrase attachment

The sentence fragment displaying ambiguity in prepositional-phrase attachment is the one for which we can easily imagine two or more different parse trees. In the timeworn linguistic examples (which are as the matter of fact hardly ever encountered in corpora) of the following kind:

Example 3. *"I saw the man with the telescope."*

Example 4. *"I saw the cat with the telescope."*

the PP "with the telescope" can attach to either the verb or the object. In the first case, the sentence without further context cannot be reliably resolved even by a human.[9] Attaching to the object noun would mean that the man is in possession of the telescope, while attaching to the verb would be interpreted as additional information about the act of seeing. In the second example, the human processor would probably conclude that the correct attachment cannot go to the object noun because cats do not usually wear telescopes. Note that in linear processing of the second example, the sentence is still ambiguous when we reach the preposition. It is only after processing the second noun that we are able to disambiguate. In this work, we say that the PP *modifies* either the verb or the object noun in order to avoid the debate about whether it is an adjunct or an argument [Schütze 1995, Merlo 2003]. To us, this question is a matter of degree rather than a categorical decision. Furthermore, the question is difficult and out of the scope of this thesis, although we are aware that the issue may play some role in PPA ambiguity resolution.

[9] Another similar example is from [Manning & Schütze 1999]: "We have not signed a settlement agreement with them."

In certain situations, knowing only the P without the second noun is sufficient to decide reliably on the correct attachment:

Example 5. *"Un juge a rejeté l'abandon de toutes les accusations /.../"*

We observe that this is so in the case of preposition "of" in English and "de" in French in direct-object constructions, because both attach to the object noun almost exclusively (we will see the exact statistics in the next chapter 3.1). Conversely, prepositions such as "into" and "despite" in English and "malgré" in French almost never have nominal attachment meanings.

Intuitively, we see that knowing the lexical identities of the preposition and the second noun helps in the attachment decision (see [Hindle & Rooth 1993, Collins & Brooks 1995] for a similar formulation).

We would like to draw the reader's attention to some other observed properties related to the study of PPA. In a situation where the object is realized as a pronoun rather than a noun, verbal attachment is much more likely:

Example 6. *"/.../ we haven't signed it with them /.../"*

Also, according to one's intuition, verbal attachments are analogous to adverbs. The set of prepositions is limited, there is a strong tie between P and N2, and only a limited set of conventionalized N2s can occur after the P. With nominal attachment, we tend to think that the choice is less restricted. In the example 7,

Example 7. *"/.../ eat a pizza with fork."*

Example 8. *"/.../ eat a pizza with anchovies."*

in order to express the instrument with which the act of eating is done, there is a restricted choice in the N2 position. We also note that in the case of verbal modification, the object noun or a noun phrase tends to get generalized. Similar observations are also shared by [Zeldes 2009, Fabre & Frérot 2002]. Another interesting property that likely influences the PPA is the use of determiners and possessive pronouns, and the distinction between definiteness/indefiniteness. Interestingly, this seems to be so for both N1 and N2, as the examples for English and French show, respectively:

Example 9 (from [Hirst 1987])**.** *"The women discussed the dogs on the beach."*

Example 10. *"The women discussed dogs on the beach."*

Example 11 (inspired by [Gala & Lafourcade 2007])**.** *"/.../ acheter des livres pour enfants."*

Example 12. *"/.../ acheter des livres pour ses enfants."*

Although the problem of PPA is generally reduced to PP occurring after direct-object transitive V-N1 constructions, the PPA ambiguity is present also in the case of indirect objects:

Example 13. *"Répondre à la crise par une nouvelle conception de l'entreprise"*

in concatenations of PPs where the attachment could go to different nouns:

Example 14 (from the Etape/Ester corpus)**.** *"/. . . / le prochain enregistrement du Masque et la plume sera cette fois en public le jeudi 28 octobre à 20h00 au studio Sacha Guitry de la Maison de Radio France pour deux emissions, /. . . /"*

with adjectival phrases [Hirst 1987]:

Example 15 (attachment either to verb or adjective)**.** *"He seemed nice to her."*

and with several verbs [Hirst 1987]:

Example 16. *"Ross said that Nadia had taken the cleaning out on Tuesday."*

With the examples above, we hope to show that the direct object PPA ambiguity, which is definitely the most studied one, is by no means the only one.

2.2.2 Approaches to disambiguation of PP-attachment

The following sections introduce possible categorizations of the work on PPA disambiguation.

2.2.2.1 Isolated vs. parsing-aware

In the majority of the NLP research done on the subject, the task usually consists of classifying quadruples in the direct transitive construction of the form V N1 P N2 as cases with either verbal or nominal attachment. The classifier's output is evaluated against the gold decisions on the same data-set, and the accuracy is usually reported. The issue of retrieving PPA cases from the raw or parsed corpora is not addressed. The question whether PPA classifiers, also called re-attachers, can actually outperform the parsers is not dealt with. We call this approach "isolated" because it does not consider PPA cases as found in the corpus or as found by the parser, but rather already extracted as an isolated set; because it does not treat the PPA disambiguation in the context of parsing. This means that the results may not be realistic and may not have practical relevance for NLP applications. The literature, both from 90's and recent, presented in the section 2.2.3, adheres to this line of research unless otherwise stated. The isolated approach was for the first time extensively criticized by [Atterer & Schütze 2007], although other research had noticed the problem already before [Foth & Menzel 2006].

We call "parsing-aware" the line of research that deliberately tried to resolve the PPA problem as a task going hand in hand with parsing. Here, the goal in general is to improve on parser's performance on the PPA cases. In addition to the work by Atterer and Schütze, the work representative of this approach is relatively recent [Agirre *et al.* 2008, Henestroza & Candito 2011, Foth & Menzel 2006].

Related to the PPA in the context of dependency parsing, we would like to point to [Bohnet & Kuhn 2012]'s observation that PPA can be particularly problematic

depending on the type of the parser employed: a transition-based dependency parser is forced to make an attachment choice at a point where only partial information about the word's own dependents is available. This can be illustrated with taking a closer look at the situation where the parser has processed the N1 in a PPA example and when it is about to consider the upcoming P (being part of a larger PP). It is forced to attach the P to either V or N1 without "knowing" anything about the identity of the P's dependent. Let us consider the following examples:

Example 17. *"Peter bought a house with an old garden."*

Example 18. *"Peter bought a house with an old friend."*

When the parser adds the arc for the P, the decision is final and the parser got only one of the above examples right. No information about the N2 was included.[10] This problem of transition-based parsing is, on the other hand, not present in graph-based parsing: the attachment site for the P is decided upon only when the N2's identity has been considered. This observation that is highly relevant in our case, would imply that, all other things being equal, a transition-based parser ought to perform worse on PPA cases than a graph-based parser. This is something we leave for future experiments.[11]

2.2.2.2 Amount of supervision

The work on PPA disambiguation can be divided into supervised and unsupervised. The supervised systems use an annotated corpus where PPA decisions are solved for training a classifier (e.g. [Collins & Brooks 1995, Hindle & Rooth 1993]). Such a corpus is, for example, the Penn Treebank. The unsupervised approach uses a corpus that was perhaps preprocessed, but does not include manual annotations of attachment decisions. Some research also focused on extracting unambiguous attachment examples from corpora by means of manually defining retrieval heuristics and then learning on these cases [Ratnaparkhi 1998, Pantel & Lin 2000].

2.2.2.3 Other divisions

It is possible to divide the prolific work on the subject according to other criteria, such as binary vs. non-binary classification ([Merlo 2003] uses 4-way classification in noun/verb adjunct or argument); the research trying to account also for PPs other than just the standard PP following the verb–direct-object pair (long PP concatenations)[12]; and exploiting information restricted to 4-tuples vs. wider context [Olteanu & Moldovan 2005] ([Altmann & Steedman 1988] even claim that for

[10] Bohnet and Kuhn propose a solution that is based on recalculation of the scores of all the histories.

[11] That is also perhaps why [Henestroza & Candito 2011] were not able to improve on the MATE parser, but suceeded with the Malt.

[12] Several PPs concatenated one after the other mean a factorial growth of possible trees/dependency graphs:
1 PP = 2 graphs

the full PP-resolution the construction of a discourse model in which the entities occurring in it are reasoned about is needed).

2.2.3 Literature survey

One of the first works on PPA disambiguation in the NLP comes from [Hindle & Rooth 1993]. They focus on studying association strength between verbs or nouns and prepositions in an unsupervised setting. They estimate the association probabilities on the 13-million word parsed Association Press corpus with the specification of what constitutes an unambiguous attachment. The probabilities are then compared based on a likelihood ratio. For testing, they create a 1000-sentence sub-corpus for which the attachment is annotated manually by the authors. In their attempt, no account is made for the N2. They achieve 80% precision and 80% recall on the V N1 P triples, compared to around 86 % precision and recall for human judges when provided only with the triples without additional context.

[Ratnaparkhi *et al.* 1994] extract supervised training material for PPA disambiguation from the Wall Street Journal part of Penn Treebank [Marcus *et al.* 1993], 20801 sentences are available for training and 3097 for testing. This is the data-set that was widely used and accepted as the de-facto standard by the subsequent research (called RRR data-set in the following). Authors' contribution is also in that they describe two different performance lower bounds: always choosing the noun attachment, which equals the nominal attachment ratio of 59% in English; and choosing most likely attachment for each preposition, 72.2%. The average human accuracy is 88.2% when provided the quadruples only, and 93.2% when given a complete sentence. With their feature-rich maximum entropy modeling, they achieve 77.7% for word-only and 81.6% for word and class feature accuracy. [Ratnaparkhi 1998] collected 910,000 unique unambiguous triples (V or N1, P, N2) from the Wall Street Journal, and proposed a probabilistic model based on cooccurrence scores calculated from the collected data. His unsupervised method achieved accuracy of 81.9%.

Also in a supervised fashion, and exploiting all four elements in a quadruple, [Brill & Resnik 1994] achieved 81.8% accuracy while employing 266 transformation rules.

[Collins & Brooks 1995] propose a disambiguation model that is similar in concept to language modeling, i.e. it employs maximal information when possible, and gradually backs off to more restricted information otherwise. Collins and Brooks use the RRR data-set and report 84.5% accuracy with morphological pre-processing. One interesting finding of theirs is that the use of low count events (mostly for the complete quadruples, which occur once or twice in the training set) contribute sig-

---

2 PPs = 6 graphs
3 PPs = 24 graphs
⋮
n PPs = $(n+1)!$ graphs
So, in a less common but nevertheless possible scenario of 5 PPs, we would thus in theory need to consider 720 different dependency graphs!

nificantly to the improvement on the test corpus.

One of the most well-known works in PPA disambiguation is without doubt the one by [Stetina & Nagao 1997], who achieve the highest results, approaching human accuracy. Their method consists of semantically annotating the corpus with WordNet concepts and then inducing a decision tree with WordNet concepts as attribute values. The motivation for including semantic classes into the disambiguation was motivated by the fact that in the work of Collins and Brooks, the precision culminated when the complete quadruple was found in the training, but these cases were very rare due to the data sparseness. Stetina and Nagao tackled this disadvantage by approximating lexical items with less sparse semantic classes. The accuracy in their experiments on the RRR data-set was 88.1%.

Stetina and Nagao were however not the only ones to utilize word senses as features in PPA disambiguation modeling. [Agirre *et al.* 2008] studied PPA disambiguation in the context of parsing (Bikel's and Charniak's parsers). WordNet semantic classes are mapped to the original tokens (nouns, verbs, adjectives and adverbs) in the subset of the Brown corpus, and the parser is then trained on the distribution of semantic classes. The evaluation is made both for parsing in general and specifically for PPA cases. With their technique, they achieved 20.5% error reduction in the PPA disambiguation task, which was also higher compared to the general parsing experiment, suggesting that lexical semantic information is particularly important in PPA resolution.[13] We can think of the incorporation of more abstract, semantic classes into parsing and PPA resolution as a way of reducing data sparseness and improving generality across domains [Coppola *et al.* 2011]. [Medimi & Bhattacharyya 2007] is another example of research using WordNet synsets as a back-off technique in the case of data sparseness. Learning only from the unambiguous examples collected from a corpus, they achieve almost 85% accuracy in PPA resolution on the RRR data-set.

[Olteanu & Moldovan 2005] take a rich-feature space (27 features) approach with the support-vector machines learning model. They explored the usefulness of manually-annotated semantic information in the form of verb-frame semantics (FrameNet) and voting with count ratios obtained from WWW to avoid data sparseness. Both types of features were found to significantly improve the overall accuracy, however the contribution is still minimal: the verb-frame feature improved the result by 1%, while querying the WWW contributed from around 1 to 2.5% in accuracy depending on the data-set.[14] Their approach is specific in that it is exploratory in

[13] [Candito & Seddah 2010] ran a similar experiment in statistical parsing of French, where terminal forms were replaced by more general symbols, particularly clusters of words obtained through unsupervised clustering. However, their experiment, while positively affecting the parsing accuracy, cannot be said to deal with semantic classes/clusters with respect to any of semantic hypotheses we made (their clusters are obtained through Brown hard clustering algorithm, which is based on maximizing the likelihood of a corpus according to a bigram language model).

[14] The exact percentage is not known as the authors only made the observation that including this type of information significantly improved the results, without explicitly exploring the actual contribution. It was also the case that the feature set combination changed while moving from no-WWW- to WWW-aided disambiguation.

nature: overall, little justification and result interpretation is provided for some of the features. It turned out that that collecting frequency counts of nominal and verbal attachments from the WWW helps overcome data sparseness problem, but this makes the approach susceptible to limitations of using WWW-aided collection of linguistic data [Kilgarriff 2007]. Other works that included WWW in PPA disambiguation, either to estimate frequency counts of quadruples directly or to estimate a co-occurrence strength, include [Kawahara & Kurohashi 2005] (87.3% accuracy on the RRR dataset with an SVM method where unambiguous examples are extracted from the corpus), [Calvo & Gelbukh 2003] for Spanish, [van Herwijnen *et al.* 2003] for Dutch, [Gala & Lafourcade 2007] for French.

[Pantel & Lin 2000] describe an algorithm for unsupervised classification of PP-attachment which uses a collocation database extracted from dependency relations and a corpus-based thesaurus that returns for a particular word a set of similar words along with similarity scores. They use frequency estimation from the corpus where possible and approximate rare events with contextually similar words (this method was not found to contribute significantly to the results). They achieve the accuracy of 84.3%. [Bharati *et al.* 2005] is an upgrade of [Pantel & Lin 2000] with a hybrid approach utilizing information from the WordNet. [Zhao & Lin 2004] is another upgrade incorporating distributional semantics. The classification decision is made according to the weighted majority vote by the k-nearest neighbours. The nearest neighbours are determined in the DSM by computing cosine similarity, Dice coefficient or Jensen-Shannon divergence on pure frequencies or mutual information values between the input vector and a training vector. The authors found out that the cosine similarity with pointwise mutual information performed best, yielding 86.5% accuracy on the RRR dataset.

Working from the generative perspective, [Toutanova 2006] achieves 85% with Bayesian networks on the RRR dataset. The goal of the paper, however, is the comparison between discriminative and generative learning, and not the PPA disambiguation. In a similar stance, [Toutanova *et al.* 2004] note (citing [Bikel 2004]) that the Collins parser uses bilexical word dependency probabilities only 1.5% time, the rest of the time backing off to one word conditioned on the POS, as a result of the data sparseness. Sparseness can be reduced with stemming, distributional similarity or WordNet, which is exactly what they show by integrating this types of information into their model. They introduce a random walk (Markov chain) model for learning the parameters for PPA disambiguation, which provides a general framework for unifying the different notions of smoothing. On the RRR dataset, their baseline model achieves 86% accuracy, while the best-configuration model (including stemming, Jenson-Shannon divergence similarity and WordNet features) achieves an impressive 87.56% accuracy. Similar results to the baseline model reported in [Toutanova *et al.* 2004] were also obtained by [Søgaard 2011] who used Bayesian networks learned with hill climbing for his generative part and conditional random fields for discriminative learning. He used only unigram features in his research (P, V, N1, N2, together with their distributional clusters).

As we mentioned in the section 2.2.2.1, [Atterer & Schütze 2007]'s paper marks

a turning point in the PPA disambiguation research, in that it provides a critique of the methodology used in the previous research. They introduce the notion of "oracle", which stands for a pre-system that extracts 4-tuples in most of the previous literature on the subject. Having an oracle thus means knowing the two alternative attachment sites. This oracle uses a gold standard corpus to extract tuples based on syntactic parses. Atterer and Schuetze remark that the performance is higher in the presence of such oracles. They argue for a new method of evaluation that does not presuppose the existance of the oracle and that is application-driven, i.e. evaluated in the context of parsing. In their experiment, they build artificial sentences of type "They V N2 P N2", taking as the data source the standard RRR dataset. They claim that state-of-the-art parsers do not differ significantly in their performance compared to PP reattachment classifiers on this task. Namely, [Olteanu & Moldovan 2005, Collins & Brooks 1995, Toutanova *et al.* 2004] (only the reattachers that do not include any additional resources such as WordNet, dictionaries, Web, named-entity recognition or stemmers, were analysed) are found to perform insignificantly differently than the Bikel's parser in one experiment where three-lexical dependencies are taken into account by the parser and in another experiment on the Penn Treebank where bi- and tri-lexical dependencies are taken into account by the parser. Importantly, this article has had consequences for how we perceive the baseline. This is no longer e.g. always choosing the nominal attachment site, but it is simply the attachment performance of the parser. A similar observation was also made by [Foth & Menzel 2006], who distinguish between the isolated and situated attachment evaluation.

For French, [Henestroza & Candito 2011] provide specialized models for parse correction in PPA and coordination attachment. They obtain the following results for PP-attachment: UAS for the baseline (parser) was ranging from 83.2 to 86.1 on the French Treebank evaluation set. With corrective models, they were able to significantly improve PPA UAS of 2 out of 4 parsers (Malt and Berkeley, but not MST and Bohnet (Mate) parser). Their approach modifies around 1–2% of all tokens, but it introduces a relatively large number of errors (changing correct parser decisions into incorrect): 29–88% of wrong-to-correct modifications are correct-to-wrong modifications. They acknowledge that a ceiling was reached and that outside resources (subcategorization for verbs, selectional preferences) should be used for this task in order to achieve better results. Their work is valuable in how they determine the set of possible heads and dependents, because they take as input the output of the parser (no gold-standard oracle). For the correction, the set of all dependents is checked in left-to-right manner, and at each dependent, candidate heads are identified in a neighborhood around the predicted head. A scoring function selects the best head. Then, the parse tree is updated accordingly. The scoring function uses discriminative linear ranking models with features encoding mostly syntactic context. They mention that PP-attachment in French accounts for around 30% of incorrect attachments and has a parser error rate of around 15%.

[Gala & Lafourcade 2007] present an ongoing research on the resolution of PPA ambiguities in French, but without providing a complete evaluation (e.g. no com-

parison to baseline). Their method consists of querying the WWW with PPA ambiguities obtained from the text parsed with the XIP parser, and learning similarities between lexical signatures from 21,048-sentence Le Monde corpus. The precision of their method ranges between 75% and 80.6%.

Some other research deserves to be mentioned. [Volk 2002] investigates the PP-disambiguation issue with the combined, supervised and unsupervised, exploratory approach on German. He recognizes the problem of manually annotated treebanks for non-English languages, which calls for exploring unsupervised methods in greater depth. His disambiguation technique relies on comparing frequencies. [Zeldes 2009] suggests an interesting, linguistically-motivated alternative using productivity measures for overcoming the problem of unseen events. Because these measures indicate how probable is encountering one-time occurrences at a certain position in the construction (or in a PPA quadruple), they could be used in PPA disambiguation (for example, verbal attacments are unlikely to have novel nouns in the N2 slot). However, this presupposes a kind of semantic disambiguation between, for example, instrumental PPs and non-instrumental PPs, because this is what influences the productivity of slots. This question is not adressed in the paper. Productivity measures were also investigated for French in [Fabre & Frérot 2002]. Their approach consists of intersecting two productivity measures, one for the number of different nouns occurring with a verb and a preposition (V+P+$|N|$), and the other for the number of different verbs occurring with a preposition and a noun ($|V|$+P+N). Their work touches upon the use of subcategorization frames. This is an interesting line of research, although not frequently encountered. [Gamallo *et al.* 2003] learn word classes from the co-specifying slots (head–modifier)[15] in subcategorization frames, and evaluate on a PPA disambiguation task for Portuguese.

[15]Their idea of co-specification is in fact quite similar to what is described in [Erk & Padó 2008]

Chapter 3

Prepositional-phrase attachment disambiguation with distributional methods

In this chapter, we present the methodology behind our experiments and the results. Before the experimental part on the disambiguation, we introduce some general statistics of the phenomenon of PPA in the corpus, we also look at the figures for the parser performance on the task. Following that, we show the functioning of our system for the retrieval of ambiguous and unambiguous PPA cases. Only then we move to the two parts of our disambiguation experiments: one including association strength in which we consider the role of the P in disambiguation; the other part including a DSM which will account for the role of both the P and the N2 in disambiguation. Our approach to PPA treatment is unsupervised in the following way: we can disambiguate given the parser output by finding cases deemed ambiguous, but without the need for gold annotations for training. We would like to point out that relying on the parser output, the task gets harder (the impact smaller), because only the cases recognized as ambiguous by the parser can be realistically expected to be checked by our PPA disambiguation techniques. This means that the recall of the true ambiguities from the corpus can be maximally as good as the parser recognizing them.

In the following, we use the term "PPA case" for any construction of a verb with direct object noun followed by a preposition and a second noun. For the sake of simplicity and to make evaluation more straightforward, we do not treat indirect objects in PPA, just like we do not treat the ambiguity of adjectival-phrase and many-verb attachments (see 2.2.1). The section 3.3.2 on retrieving includes more precise rules for what constitutes a PPA case.

3.1 Quantitative analysis of the problem

3.2 Experimental design and goal

We use the French Treebank (FTB) [Abeillé & Barrier 2004] for testing in our experiments. It contains 12351 sentences, and was created from the newspaper Le Monde. More specifically, we use the dependency converted tree-bank described by [Candito *et al.* 2010].

For learning selectional preferences and distributional semantic models, we use the French Gigaword 2nd edition corpus containing news-wire text data from Agence France Press and Associated Press French Service [Ângelo Mendonça *et al.* 2009]. A part of the corpus was tagged with the TreeTagger and parsed with MATE. It contains 36,355,130 sentences and 477,534,407 words.

The MATE parser was used for parsing the Gigaword corpus and for experiments on the FTB. The models for the parser were trained on the FTB. On the test set, the MATE parser obtains UAS of around 87%.

3.3 Preliminary experiments

3.3.1 Error analysis of the parser output

We performed an error analysis on the development set of 120 sentences of the FTB, which shows the performance of the parser on the PPA cases (as defined above) in the lower part of the table. The upper part introduces gold-annotation statistics on the same set of sentences.

Sentences	120
PPA cases with V or N1 att.	79
PPA per sent.	1 per 1.39
$\frac{\text{verbal}}{\text{nominal}}$ att. ratio	0.44
De-only cases	37
De-only cases: $\frac{\text{verbal}}{\text{nominal}}$ att. r.	0.054
Non-de cases	42
Non-de cases: $\frac{\text{verbal}}{\text{nominal}}$ att. r.	0.786
Complex prep. cases	5
Parser ER	0.19
Parser ER "de"-only cases	0.054
Parser ER non-"de" cases	0.31
Parser ER complex prep. cases	0.2
Parser ER on gold nominal att.	0.11
Parser ER on gold verbal att.	0.286
*PPA cases with another att.	26

Table 3.1: PPA statistics for the gold decisions and the parser's output on the development corpus

The occurrence rate of the PPA is 1 per 1.4 sentence,[1] and the attachments are slightly more frequently nominal (66%) than verbal (44%), which is undoubtedly a consequence of the most common French preposition "de". This preposition thus occurs in 47% of all PPA cases and is attached to the object noun in 94.6% of time. If we discard "de", the ratio between verbal and nominal attachment changes to 79%

[1] This is likely to vary with respect to how the "PPA case" is defined.

in favor of verbal attachments.[2] Other figures from the table show: the number of complex prepositions (5), where the complex preposition ("locution prépositive") is any preposition constituted of more than one word ("quant à", "au niveau de" etc.); number of PPA cases with attachments other than to V or N1 (mostly this is the situation with several PPs following one after the other).

The general error rate for the parser on the PPA cases as defined above is 0.19 (±0.07 with a 95% CI). The parser error rate is the highest on the non-"de" cases: 0.31 (±0.083 with a 95% CI). Then, we also know that the parser ER is higher for the attachments that are verbal in gold. Since the ER on verbal attachments is higher than on nominal attachments, we can say that the parser is somewhat biased towards nominal attachments.

Based on the observations in this section, we decide to only consider non-"de" PPA cases in our experiments. Firstly, this could increase the impact of our disambiguation system, and secondly, because "de"-attachments are so common and display a very low parser error rate, it does not make sense to include these cases into disambiguation.

3.3.2 Retrieval of PP-attachment cases

We implement a retrieval system that outputs a list of lemma quadruples of the form <V N1 P N2> (optionally with the attachment decision of the parser or the gold standard) given as input a parsed corpus in the CONLL format[3].

The retrieval script relies both on POS-tag and dependency relations. This is in line with the research such as [Atterer & Schütze 2007]. We also complement these two types of information with a lexicon of complex prepositions.[4] This is needed because of the frequent mistagging of complex prepositions. In this way, preposition occurrences are determined also by lexicon look-up.

In our retrieval, we linearly process the texts in the corpus and extract a quadruple if:

- P is tagged as preposition or is in the lexicon, and is not a form of "de"
- P head is in "obj" relation with N2 dependent
- N1 is a common noun, a proper name or an abbreviation, and it is in "obj" relation with V head which must have a POS-tag of a verb
- P dependent is in some dependency relation with either V head or N1 head

We would like to emphasize that in the current version of the retrieval, PP concatenations are also extracted, but they are represented as a V N1 P N. This

[2] In an analysis before the one described here, we calculated the statistics on 65 sentences from the test sub-corpus. The verbal attachment for "de" was the same (4.5%), while the parser error rate on the verbal attachments was even more elevated (50%).

[3] `http://ilk.uvt.nl/conll/index.html#dataformat`

[4] The lexicon is partly built of complex preposition lists from various linguistic repositories on the web, and partly of complex prepositions extracted from the FTB.

means that having a sentence fragment with the abstraction V N1 P1 N2 P2 N3, the following two representations will be created: V N1 P1 N2 and V N1 P2 N3. This is not optimal, however, since it is impossible to arrive at the attachment of P2 to the preceding PP (N2), which is perhaps less often but still encountered in the corpus. An improved version should either not consider the concatenated PPs at all, or build alternative tuple representations.

Total sentences (dev subset)	120
Number of gold non-"de" cases	42
Number of retrieved non-"de"	35
Precision	0.886
Recall	0.738

Table 3.2: Retrieval results

As the above table shows, we can extract 35 non-"de" cases from 120 sentences in which there are 42 true non-"de" cases. The precision (here measured as the percentage of retrieved instances that are true PPA instances in the gold) in a real-case scenario is then 0.886 $\pm 0.057_{95\% CI}$, while the recall (the percentage of true gold PPA instances retrieved by the system) is 0.738 $\pm 0.079_{95\% CI}$. Precision is for example lowered by the parser attaching to the wrong verb. The recall is lowered by the following:

- restrictions in our retrieval system that, for example, demand the object to have a POS-tag for noun, but because of mistagging it was not retrieved
- parser's attachment is not to V nor N1, while the gold attachment is
- N1 is attached to a coordinating conjunction instead of a verb
- complex prepositions are sometimes not tokenized as one unit: a pre-processing of the complete corpus would be necessary in order to remedy this

On this particular data-set and taking into account the retrieval quality, what would then be the maximal contribution we could expect from the disambiguation system? Since we retrieve 35 cases, we can try to correct parser's decision in 35 cases. But 4 cases were not truly ambiguous – in gold, the PP was attached to neither V nor N1 – (lower precision), so we can only hope to correct attachment site in 31 cases. Of course, the majority were already correctly attached by the parser. Precisely, for our 120-sent. development set, 8 cases were falsely attached by the parser. Without generalizing from this particular analysis based on a particular (relatively small) corpus, the impact of our correction system would be limited to around 8 non-"de" cases per 120 sentences (i.e. 1 case per 15 sentences). Thus, given parser's performance and the retrieval quality, a disambiguation system's impact could be maximally 1 false-to-correct modification of a non-"de" case per 15 sentences.

3.4 Prepositional co-occurrence model

3.4.1 Experiments

In our first disambiguation experiment, we explore the effect of association strength between the V or N1 and the P. We extract from the Gigaword corpus the V+P and N+P lemma pairs by defining rules employing both POS-tags and dependency relations. We create two different collections: in one, the pairs are extracted regardless of the surrounding context, while in the other, the pairs are extracted from the so-called unambiguous contexts. With unambiguous contexts, we want to be (more) sure that the P in the pair really attaches to the first element in the pair, so we define unambiguous contexts in which alternative attachment is not possible. This is done by simple heuristics, inspired by [Ratnaparkhi 1998], such as:

- unambiguous nominal attachment (noun phrases in the beginning of a sentence): no V should occur between the start of the sentence and the P
- unambiguous verbal attachment: no N should occur between the V and the P

We thus obtain 25,437,541 V+P and N+P pairs for ambiguous and 18,999,728 pairs for unambiguous contexts (i.e. 25% reduction in the size of the ambiguous collection).

We use mutual information as a measure of association strength. The incorporation of MI into attachment resolution can be justified as described in the following. If we think of our task as classification, then the objective is:

$$A \in \{nom, ver\} \tag{3.1}$$

$$\arg\max_A P(A|v, n1, p, n2) \tag{3.2}$$

The term above can be decomposed:

$$\arg\max_A P(v, n1, p, n2|A)P(A) \tag{3.3}$$

$$\arg\max_A P(v, n1, p|A)P(n2|A)P(A) \tag{3.4}$$

$$\arg\max_A P(v, n1|p, A)P(p|A)P(n2|A)P(A) \tag{3.5}$$

The term $P(v, n1|p, A)$ can be directly estimated with MI. If A is verbal, then the term can be rewritten as $P(v|p)P(n1)$, and if A is nominal, then $P(n1|p)P(v)$. The PMI, as defined in the equation 2.2, is calculated with the formula $log_2 \frac{P(w_1,w_2)}{P(w_1)P(w_2)}$. In our case, we are interested in the maximum value, so attachment is verbal if:

$$\frac{log_2 \frac{P(v,p)}{P(v)P(p)}}{log_2 \frac{P(n1,p)}{P(n1)P(p)}} > 1 \tag{3.6}$$

This fraction can be shown to correspond to the terms $P(v|p)P(n1)$ and $P(n1|p)P(v)$:

$$\frac{log_2 \frac{P(v|p)P(p)}{P(v)P(p)}}{log_2 \frac{P(n1|p)P(p)}{P(n1)P(p)}} = \frac{log_2 \frac{P(v|p)}{P(v)}}{log_2 \frac{P(n1|p)}{P(n1)}} = log_2 \frac{P(v|p)P(n1)}{P(v)P(n1|p)} \tag{3.7}$$

If this was our model, all parts of the equation 3.5 would need to be estimated. For example, P(A) could be the prior of 0.786 in the case of verbal attachment. The other two terms would also be priors: one for a specific preposition and the other for a specific N2.

Because we know that the attachments are mostly verbal, the task can be thought of as detection, in which the default attachment is verbal, and we detect a positive instance (i.e. nominal attachment) if we have sufficient evidence, meaning that the ratio between two scores should be above a certain threshold. Thus, the detection can be viewed as a trade-off between failing to choose the nominal attachment where the nominal attachment is correct, and choosing the nominal attachment when verbal is correct. Thus, for this experiment and the one in the next section, we evaluate by looking at the full spectrum of detection results for different thresholds, which is more meaningful than just reporting one figure (say, accuracy) for one threshold. We show the results by plotting either a ROC curve or a precision-recall (PR) curve [Davis & Goadrich 2006]. Before defining what the curves represent, we introduce some basic terminology. A positive example is the detection of a nominal attachment in our case. A negative example is not detecting anything and leaving the attachment as verbal. True positives (TP) are then examples correctly detected as positives. False positives (FP) refer to negative examples incorrectly detected as positive. True negatives (TN) correspond to negatives correctly detected as negative. And finally, false negatives (FN) refer to positive examples incorrectly detected as negative. Precision is the ratio $\frac{TP}{TP+FP}$, recall is the ratio $\frac{TP}{TP+FN}$, and the False positive rate (FPR) is $\frac{FP}{FP+TN}$.

The ROC curve plots the FPR on the x-axis and recall on the y-axis. The PR curve plots the recall on the x-axis and precision on the y-axis. The more the ROC curve approaches the left side and then the upper left corner of the plotting area, the better the system. The line connecting the bottom left corner with the upper right one is the performance of a random detector. For the PR curve, the more it approaches the upper right-hand corner, the better the system. Both curves are in fact very similar. Each point in ROC or PR space represents a specific detection system, with a threshold for calling an example positive. It is sometimes claimed that PR curves can expose differences that are hardly observable in the ROC space. We will usually report the statistical significance values (p-value) and average differences (d-value) obtained by a paired permutation significance test for comparing Area under curve (AUC)[5] for two detectors.

[5]AUC is equal to the probability that a detector will rank a randomly chosen positive example higher than a randomly chosen negative one.

The goal of the experiment is to find out whether stronger association between the elements V and P results in a more likely verbal attachment than nominal, and vice versa. We present our results from calculating with the PPMI and with the LPMI. In both experiments, only the co-occurrences with the frequency of > 5 are taken into account. We test the technique on the corpus of 3,398 non-"de" quadruples extracted from the FTB. However, for the cases where both MI-derived values (V+P and N1+P) are zero or non-existent, we do not want to force any decision, so we eliminate these instances from the experiment.[6] We are thus left with 2460 instances for the ambiguous context configuration and 2368 for the unambiguous one.

The following plot shows the results of the LPMI model compared to the PPMI model:

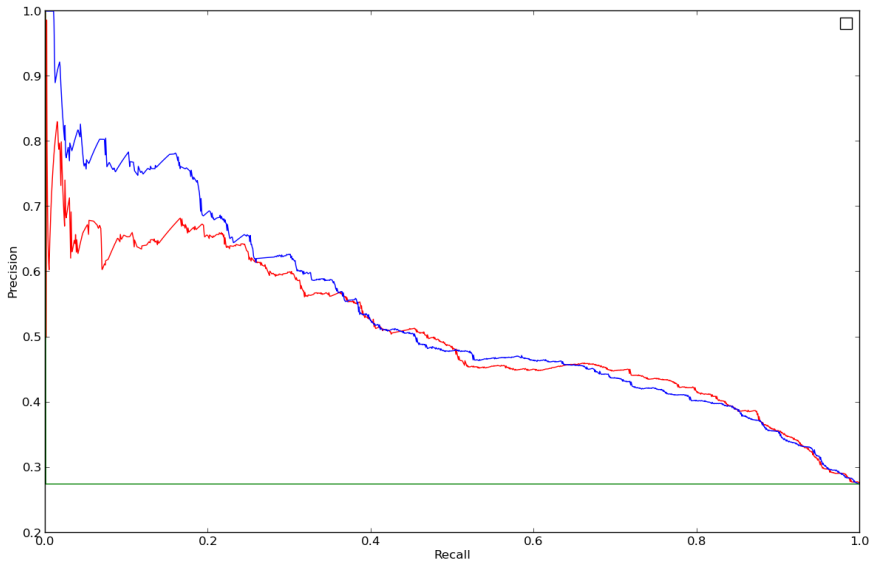


Figure 3.1: PR curve for the prepositional co-occurrence model with PPMI weights (red) or with LPMI weights (blue), obtained from ambiguous contexts. The baseline of always choosing the verbal attachment is the line in green. Number of instances = 2460.

Both models perform significantly better than the baseline of always choosing the most common, verbal attachment (of course, such a model never detects any nominal attachment). The results of the statistical significance test are almost identical for both measures: d=0.241, AUC= 0.74, p=0. The overall difference between both models is small and not significant. However, we can observe slightly different behaviour for both models, namely the models' precision behaves very differently

[6] For the cases where one of the values is zero, we set the value to a very small positive values, essentially zero, that allows calculating the ratio.

depending on the recall: LPMI outperforms PPMI at low recalls, while at recall values of around 0.7 and more, the PPMI achieves slightly better precision than the LPMI. For a conservative detection, where the difference between the models is most distinctive, we are inclined to say that it is the weighting of the MI score by the joint probability of a co-occurrence event (LPMI) that has a positive effect, thus contributing to a more precise detection.

Even though the precision at high recall looks quite small for both measures (ranging from 0.47 at the recall of 0.7 to a smaller precision at higher recall), one should be aware that this figure is lower because it only indicates how precisely nominal attachments are detected, not how precisely we choose attachment sites in general.

Next, we are interested in the contribution of the unambiguous context in the construction of the co-occurrence models. Once again, we plot PR curves for both LPMI and PPMI models.

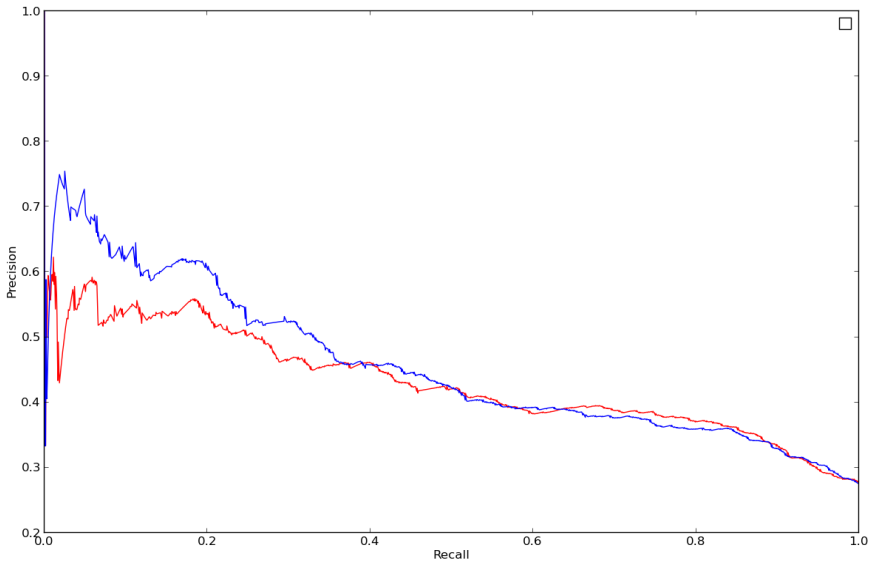


Figure 3.2: PR curve for the prepositional co-occurrence model with PPMI weights (red) or with LPMI weights (blue), obtained from unambiguous contexts. Number of instances = 2368.

Overall, we see that the detection quality decreases compared to the setting with ambiguous contexts. This goes against our expectation. A possible reason could lie in the rules for defining unambiguous contexts, which also use dependency relations. The parser's accuracy on the Gigaword corpus is lower than on the FTB, a fact mostly due to errors in sentence segmentation.

When observing the full spectrum of values on the plot, we see that the difference between the LPMI and the PPMI follows the same pattern here as in the previous plot. This suggests that these differences are due to the way the LPMI and the PPMI are calculated rather than due to the characteristics of the data.

The performance in this experiment is still significantly better than the baseline for both models (PPMI: d=0.187, AUC=0.688, p=0; LPMI: d=0.188, AUC=0.687, p=0). The LPMI model from unambiguous contexts performs significantly worse than the one from ambiguous contexts: d=0.199, p=0. In the case of PPMI, the results from the statistical significance test are: d=0.198, p=0.

3.4.2 Summary

We saw in this section that a partial probabilistic model, only accounting for either V or N1 and the P, can provide us with quite useful information, despite its simplicity. It is, however, affected by the size of the corpus used for learning word pair probabilities. In our attempt of learning the model from unambiguous contexts, the results are, counter-intuitively, even worse than from ambiguous, undefined contexts. In more general terms, we can note that in this experiment, the support for one attachment or the other comes from the preposition co-occurring with one of the preceding elements more prominently than with the other.

3.5 Vector space model

3.5.1 Experiments

In these experiments, we want to test our intuition that the semantic similarity between N1 and the PP, compared to the one between V and PP, facilitates the attachment decision. When the semantic similarity between N1 and PP is higher than the one between V and PP, we decide that the attachment should be nominal. In one experiment, we take into account only the N2 from the PP, and in another experiment, we let the PP be represented by a composed vector of P and N2.

We build a word-context matrix of 2,816 rows and 10,000 dimensions. Rows are the elements of the quadruples extracted from the FTB (V, N1, P, N2). For convenience, we call them "terms". Dimensions are the 10,000 most frequent words from the Gigaword corpus. The least frequent context word has the frequency of 1411. Because of the constraints on both rows and columns, we reduce the matrix to a size that can be handled in the working memory (225MB) without problems, and at the same time we reduce the sparsity of the values. The matrix thus contains 28,160,000 elements, of which only 74.1% are zeros.

For both terms and context words, we use lemmas. Our base DSM is a simple frequency count matrix, constructed by scanning through the Gigaword corpus and noting all occurrences of a term with the context words in a window of size 3. The context word can be any word satisfying the POS constraints (we forbid some POS such as pronouns and determiners). The context window is implemented as a flexible

window of varying size with the maximum of 3 words to the left and 3 words to the right of the term. So, if our term is the first word in the sentence, only the 3 words right to it will be checked and noted (if they are true context words). If the term is the second word in the sentence, one word to the left and 3 words to the right will be recorded (if the condition just mentioned is met), etc.

We experiment with four different weights for our matrix: the logarithm of the frequency count of each matrix element (see equation 2.1); the PMI, LPMI and PPMI variants of the mutual information (see equations 2.2 and 2.5).

In addition to the dimensionality reduction on the basis of the frequency of context words, we use the SVD technique to obtain a reduced matrix (see figure 2.4). In our case, we opt for 300 dimensions in the new truncated matrix. This means that we use 300 σs (singular values) in the reconstruction of the new matrix. The number of dimensions chosen is motivated by the frequent use in the literature (e.g. [Baroni & Zamparelli 2010] reduce a 12,000 by 10,000 matrix to 12,000 by 300), but also by the fact that in this way, we account for most of the variance in the original data, that is 92.1%.[7]. By applying the SVD in the experiments, we are less concerned by the practical, size-related advantages of the SVD. What is of interest is obtaining a DSM that will better represent the semantic similarity between the items compared, thus resulting in improved PPA disambiguation.

In our work, we use exclusively the cosine as the metric for vector comparison (equation 2.6). The cosine is calculated between two vectors, which provides a score ranging between 0 (if none of the matrix cells were negative; e.g. in a plain frequency DSM) or -1 (if some values were negative; consider the case of a singular-value decomposed DSM) and 1 (perfectly aligned vectors). We normally perform two cosine calculations (e.g. between N1 and N2, and between V and N2) in order to get a score by division. Intuitively, this score would represent that two cosine-compared vectors are more similar (two terms are more semantically similar) than the other two, and we can think of the final score resulting from division as the amount of confidence that two vectors are more similar than the other two. It is the higher semantic similarity between the elements in the quadruple that is expected to correlate with the increased possibility for a particular attachment. We are thus interested in observing thresholds δ (or the trade-off) for the detection of a nominal attachment based on the ratio. To illustrate our hypothesis, we take a N1 N2 pair and a V N2 pair from the quadruple V N1 P N2. Then, when one pair is more semantically similar than the other, we simply consider this as support for deciding on the attachment. An increased semantic similarity of N1 N2 over the pair V N2 thus provides us with the support that the attachment is nominal.

[7]Note that the σs are ordered from the highest to the lowest, and that each of them represents the variance it is able to capture. For example, reducing our matrix to only 2 dimensions would mean accounting for 35% of the variance in the original data. The singular values are distributed in the form of a Zipfian curve: very few values are needed to account for a large amount of variance, and the contribution from the tail of the curve/distribution is negligible.

$$A_{nom} \text{ if } \frac{Cos(n1, n2)}{Cos(v, n2)} > \delta \tag{3.8}$$

In our first attempt, we only account for the role of the second noun in the attachment resolution. However, we can also try to incorporate the P in our semantic comparisons, thus arriving at a complete PP representation which is then compared to the V and the N1. We obtain the PP by composing individual vectors for the P and the N2. We define composition as vector addition and vector multiplication only [Mitchell & Lapata 2008]:

$$pp = \alpha p + \beta n2 \tag{3.9}$$

$$pp = p \cdot n2 \tag{3.10}$$

A composed vector representing the PP can thus be either a result of adding the vectors for the P and the N2, where α and β are arbitrary weights, or a result of multiplying vectors. In the experiment incorporating the complete PP, our task becomes the following:

$$A_{nom} \text{ if } \frac{Cos(n1, f(p, n2))}{Cos(v, f(p, n2))} > \delta, \text{ where } f \in \{addition, multiplication\} \tag{3.11}$$

All the experiments presented in the continuation are tested against the complete FTB. In the first experimental setting, we start with a DSM with plain frequencies as values. At this stage, we do not consider vector composition, so we only perform comparisons with the N2 alone. We compare this DSM with two other DSMs, one with removed 100 most frequent context words (this way, we eliminate the function words that incidentally passed the POS filtering on the context words because of inaccurate tagging), another where plain frequencies are transformed with the logarithm. When a significance test is carried out, we find out that DSMs perform poorly, and that none of the DSMs perform better than the "random" curve, which is the baseline of always choosing the verbal attachment. P-values for the three models range between 0.1 (the plain-frequency DSM) and 0.8. We deliberately avoid interpreting any differences between them because none performed significantly better than the baseline.

In the second step, we apply the mutual information weights to the plain-frequency DSM. Here, we can see the improvement over the baseline for LPMI (d=-0.0219, AUC=0.478, p=0.037) and PPMI (d=0.033, AUC=0.533, p=0.008), but not for the plain PMI. The plot below shows the performance at various thresholds:

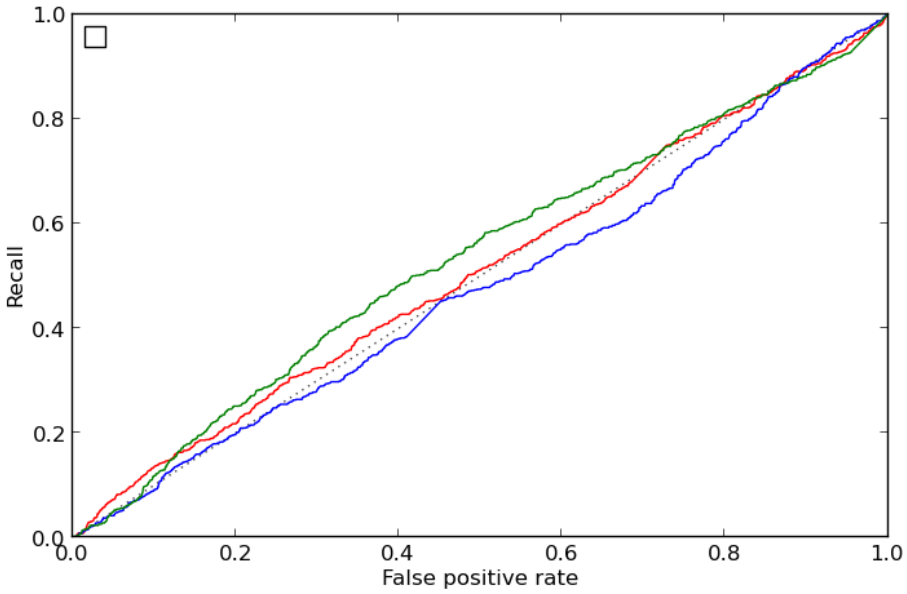


Figure 3.3: ROC curve displaying the PMI-weighted DSM (red), the LPMI-weighted DSM (blue) and the PPMI-weighted DSM (green).

We see that the performance of the LPMI-weighted DSM is significantly worse than the baseline. This simply indicates that the LPMI does contain some useful information, but is applied in an inappropriate way. As a side experiment for the LPMI-weighted DSM, we only based our detection on the semantic similarity between N1 and N2, without calculating the ratio (meaning without using the similarity between V and N2). In this way, LPMI did perform significantly better than the baseline, implying that the information it obtained from the similarity between V and N2 did not contribute to better results, but actually severely degraded them. To sum up, our findings here confirm the observations from the literature that PPMI performs better than the plain PMI (the difference is statistically significant at d=-0.0223, p=0).

Having seen the influence of the weighting schemes on the detection results, we now turn to investigating the effects of dimensionality reduction. We reduce the number of dimensions of the best-performing DSM configuration from the last experiment (the PPMI-weighted DSM) to 300 and to 2 by applying the SVD. Consulting the ROC curve reveals that the DSM with 300 dimensions outperforms the 2-dimensional DSM, and probably also the PPMI-weighted DSM from the previous experiment. Compared to the baseline, the difference is statistically significant at d=0.0676, AUC=0.568, p=0.

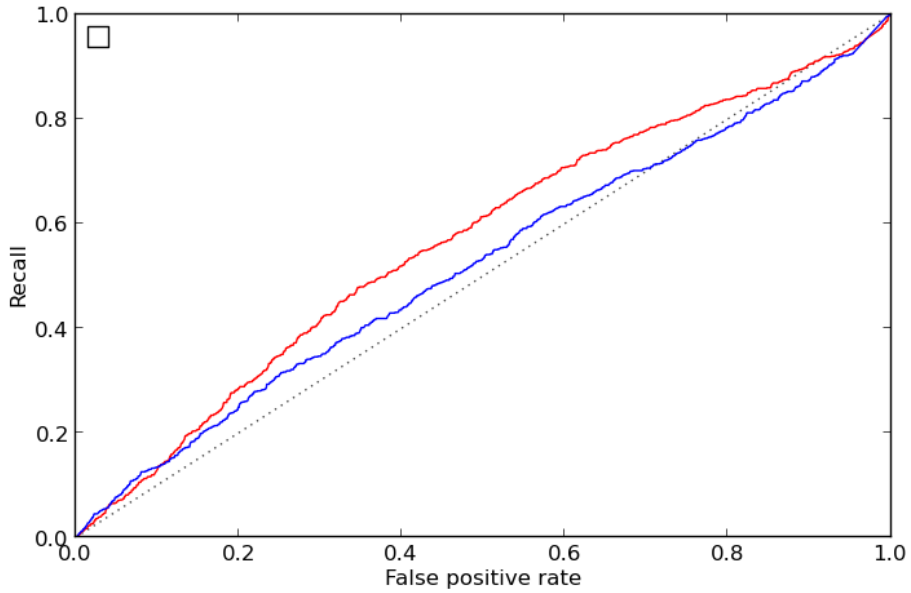


Figure 3.4: ROC curve displaying the PPMI-weighted DSM after truncation with the SVD to 300 dimensions (red) and 2 dimensions (blue).

Compared to the 300-dim. DSM, the 2-dim. DSM results in a worse performance. The difference is statistically significant at d=0.0459, p=0 (AUC for the 2-dim. DSM is 0.522). This can be explained by the fact that the 2-dim. DSM suffers from the loss of variance (remember that it only captures 35% of the variance in the original DSM). In order to be sure that the application of SVD brings about an improvement over the PPMI-weighted DSM without dimensionality reduction, we test for the significance of difference between the two: d=-0.0346, p=0.

In the next experiments, we try to incorporate fuller information about the PP by composing the N2 with the P. We thus measure the similarity between V or N1 and the vector resulting from the composition of P and N2. If we use the addition with equal weights (α=β=0.5) as the composition function, and use the best possible parametrization from the last experiments (300-dimensions, PPMI), we obtain a detection curve that performs better than the detector that only uses the N2 information: d=-0.0128, AUC=0.58, p=0. On the plot, we see that the improvement at the area of interest (where the FPR is low) is greater than for mid-ranged false-positive rates. However, the improvement in general looks quite small.

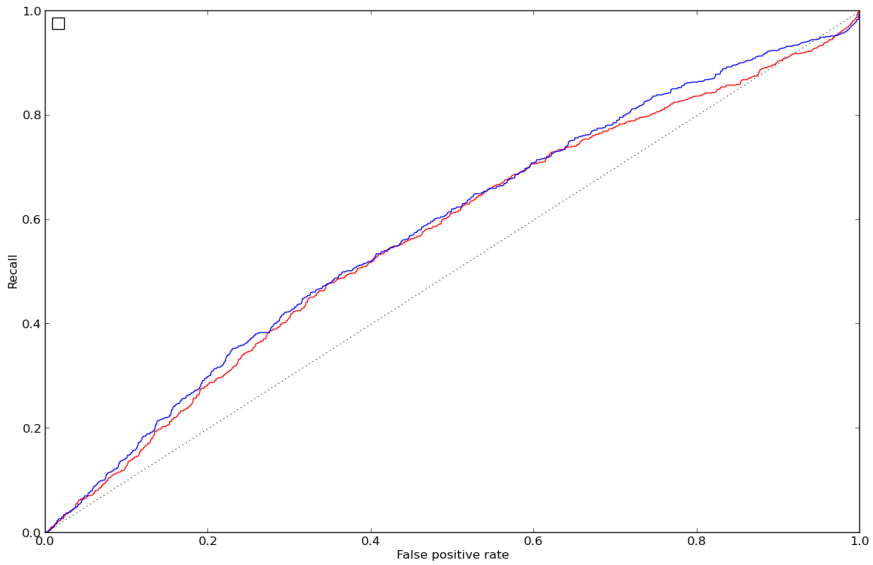


Figure 3.5: ROC curve displaying the PPMI-weighted DSM after truncation with the SVD to 300 dimensions, with the composed PP representation (blue) and without it (red).

We also tried to tune the weights in the addition function, but no setting could significantly improve the default weight of 0.5. And although the multiplication was expected to outperform addition, as is normally accepted in the research on composition, this was not so in our experiment. The DSM with the PP representation obtained by multiplication did not perform better than the baseline.

In a further exploration with respect to distributional semantic composition, it could be argued that the preposition should not be composed only with the N2, but that it should be integrated with the V or the N1 as well, since in some situations P can be more easily thought of "belonging" to the verb than to the N2 (e.g. "réflechir à/sur"). The decision on which parts of the ambiguous quadruple should be composed could be motivated by the PP type (argument or adjunct), or the sub-categorization frames, as long as these types or frames could be learned automatically. We received some justification for this line of thought by conducting an additional experiment, in which we compose the P *both* with the N1/V and the N2, that is, without any attempt to discriminate between the situations in which the P would be better composed with the N1/V and the situations in which the P would be better composed with the N2. Even though such an experimental setting is purely exploratory, the findings are promising. Such a DSM is parametrized with 300-dimension SVD, PPMI and the addition as composition function for both V/N1+P and P+N2 (weight set to 0.5).

$$A_{nom} \text{ if } \frac{Cos(\alpha n1 + \beta p, \alpha p + \beta n2)}{Cos(\alpha v + \beta p, \alpha p + \beta n2)} > \delta, \text{ where } \alpha = \beta = 0.5 \tag{3.12}$$

The DSM outperforms the best previous configuration (300-dimension SVD, PPMI, composition of the PP-only by addition with weight of 0.5) with the following statistical summary: d=-0.0111, AUC=0.591, p=0. The research in this direction is in our opinion certainly worth exploring in more detail.

3.5.1.1 Integration with the parser

In this experiment, we try to position the PPA ambiguity resolution by means of DSM-based detection into the context of parsing. The MATE parser trained on the FTB, provides us with the baseline UAS of 86.93%. In the experiment, we use a modified version of the MATE parser that is forced to keep the preannotated dependencies while linking remaining unattached words. We call this type of parsing constrained parsing. This is a novel technique that guarantees that the resulting dependency tree is optimal.[8] We thus incorporate attachment decisions as pre-annotations for use in constrained parsing. Before the parser processes the test part of the FTB, we annotate the attachments (we build the corresponding dependency arcs) according to the semantic similarities between the V/N1 and the N2. These are acquired from the DSM which uses PPMI-weighted elements and which is truncated to 300 dimensions with the SVD. The parser is then run on the pre-annotated test corpus, and the UAS is calculated. The following table provides the overview of the results for different thresholds used in detecting nominal attachments. In order to determine the maximum impact we can expect from the PPA disambiguation, we also force the correct attachments (dependencies) for all the quadruples recognized by our retrieval system. This oracle achieves the UAS of 87.25%. Note that we do not report the LAS because the DSM-driven disambiguation cannot provide any information about the labels of the relations.

Threshold	Deps pre-annotated	UAS
0.0462	62	0.863
0.5326	62	0.865
1.0190	62	0.868
1.5055	62	0.871
1.9919	62	0.8726***
2.4784	62	0.8726***
2.9648	62	0.8722

Table 3.3: Parsing experiment with DSM-based attachments. *** represents the best result

[8] A simple corrective model could modify the parser's output, but the dependencies that were not modified would not be optimal anymore from the parser's perspective.

The thresholds 1.9919 and 2.4784 produce the highest UAS of 0.8726%, which is even slightly higher than the oracle performance. This apparent paradox can be explained by the fact that the parser makes more mistakes *outside* the quadruples when it uses the oracle pre-annotated attachments than when it uses only some correctly annotated attachments. Even though we observe a small increase in the results when DSM-based attachments are used, this difference is not statistically significant.

In table 3.3, all suggestions by our detector were taken into account when preannotating the test corpus before constrained parsing. However, as mentioned before, the cosine ratio can be considered as a measure of the confidence for the propositions of our DSM-driven detector. Hence, given the already good performances of the parser alone, a better integration between our detector and the parser can be realized by constraining the parser to keep only the most reliable dependencies suggested by our detector. We test now such an experimental setup by using two thresholds: one for the most likely (in the sense of the cosine ratio) nominal attachments cosine r. $> \delta_{nom}$, and another for the most likely verbal attachments cosine r. $< \delta_{ver}$. By varying both δ_{nom} and δ_{ver} over the full range of observed cosine ratios, we actually observe that there are many possible values for both thresholds for which the DSM-driven detector corrects the PPA proposed by the parser.

We report next an excerpt of these thresholds, along with the absolute number of correct attachments made both by the parser on the PP-attachment cases proposed by the DSM detector and the DSM detector.

	δ_{ver}	δ_{nom}	Number of attachment cases	Correct att. by the parser	Correct att. by the DSM-driven detector
	1.078217	1.7003012	44	31	33
	1.078217	3.2809374	39	26	32
	1.078217	1.3897803	46	32	34
	1.078217	1.9061964	43	30	33
	0.9382211	2.546115	36	23	27
	0.9382211	1.5775068	39	25	28
	0.9382211	2.680369	34	21	26
	0.9382211	1.3190644	41	26	28
	⋮	⋮	⋮	⋮	⋮
Avg. accuracy				0.69	0.769

Table 3.4: Parsing experiment with DSM-based attachments for 483 threshold combinations (onyl an excerpt is shown together with the average accuracy).

We can observe on this excerpt that the DSM detector often proves useful to

correct the parser decision, and also that exploiting the confidence measure of the cosine ratio helps in focusing on the most useful suggestions of the DSM-driven detector.

Although the absolute number of occurrences of dependencies corrected by the DSM detector is too small on this test corpus to result in a significant increase of the global UAS, this integrated model is very interesting, because it exploits the best of both the parser and the DSM detector, in this way always providing a small but consistent improvement of the parsing accuracy for the PP-attachment. Such a solution may prove especially powerful in semi-supervised learning approaches, which need to integrate additional information that is not already modeled in the parser itself to parse a very large corpus, and iteratively retrain new versions of the parser that better integrate this new type of information.

3.5.2 Summary

In this section, we implemented a system for PPA resolution that is able to perform better than the baseline model of always choosing the most frequent, verbal attachment. The system is a DSM weighted by PPMI, truncated to 300 dimensions by the SVD, and incorporating composition of vectors for the preposition and the second noun, in order to arrive at a single semantic representation for the PP. Despite the fact that our DSM configuration performs significantly better than the baseline, the detection, as observed from the ROC curves, is in practice not very accurate. We would normally expect to see a well-defined trade-off between precision and recall, or between recall and the false positive rate, however, we cannot observe it from our data. This can simply be caused by the fact that the information we can get from the second noun (or from the second noun in combination with the verb or the first noun) helps in some cases, but often, even if the semantic similarity between, say, the first and second noun is very high (and the similarity between verb and the second noun is low), the attachment is still, counter-intuitively, verbal. This could then be explained by at least three things: the problem of PPA consists of different types of constructions/phenomena where sub-categorization frames for verbs, for example, could play a role; the intuition that the choice of attachment site is affected by the semantic similarity, as defined in this section, is not reliable or is not the only predictor; errors from the retrieval of PPA cases, POS-tagging and parsing contribute noise in our distributional semantic models.

The above observations suggest that the semantic similarities we obtain from a DSM could be used as a supplementary source of information for a strong base model, perhaps the prepositional co-occurrence model, or they could be integrated in the parsing. Indeed, our last experiments tried to situate the DSM-driven detector in the context of parsing by pre-annotating the dependencies. We use a novel approach in which the best of both worlds, the DSM-obtained attachments and the parsing model, can be integrated in an optimal result. Even though it was not possible to significantly improve the UAS of the baseline parser, this is largely understandable, since the problem (or our definition of it) is quite specific and narrow. We did

find out, however, that with distributional semantic modeling we are able to resolve many cases that were incorrectly attached by the parser.

A note on the experiment implementation

The code for the experiments was written in Python and is being made available in the public repository `https://github.com/SimonSuster/PP-attachment-disambiguation`. We made extensive use of the following packages: yard (for plotting ROC and precision-recall curves), numpy and scipy (for scientific computations on arrays and matrices), aima (for co-occurrence counts), myconllutils (to manipulate CONLL format files). Some plots were produced in R.

Chapter 4

Conclusion

In this thesis, we focused on the role of distributional methods, especially distributional semantics, in the prepositional-phrase attachment ambiguity resolution. We investigated both the contribution of a co-occurrence-based model, accounting for the preposition along with the either attachment site, and a word space model, which provides a more complete representation of the entire prepositional phrase by measuring semantic similarity between items in an ambiguous case. We defined the task of PP-attachment resolution as a detection, with the information obtained from our models acting as a confidence that the attachment is of a particular type. Positive and encouraging results were obtained with both approaches, the co-occurrence model and the distributional semantic model. The latter was integrated into parsing by providing dependency annotations which were then taken into account, but not altered, by the parser. Even though the contribution of the detector on our test set proved too small to observe a significant improvement in the overall parser performance, we saw that the attachments proposed by our method outperform a large number of attachments annotated by the parser, and that this approach is particulary appealing for consideration in the future.

Further work would need to investigate the following points in order to reach a more complete understanding of the role of distributional semantic information in PPA and more broadly in structural ambiguity resolution:

- Exploring settings with chances for larger impact, i.e. where the contribution from a DSM could be of most value, such as parsing spoken corpora or unsupervised parsing.

- The same approach that was presented in this thesis should be tested on another language, perhaps starting with English, because this would make comparisons to other results in the research community easier and more effective.

- Exploring compositionality in a greater degree. We saw that whenever we perform composition, we obtain more meaningful semantic vectors, at least for our application of PPA disambiguation.

- Incorporating parsed-text to a larger extent in the DSM. The problem we faced that hindered such an attempt was the segmentation errors on the Gigaword French corpus, which makes the output after parsing more affected by the errors.

- Aiding the parser can be implemented in several ways. One option, as presented here, is the pre-annotation together with constrained parsing. Another solution would integrate with the parsing feature scheme, which is also attractive as it tries, similarly, to arrive at the optimal result from the point of view of the parser. Yet another possibility is processing raw text with semantic classes which are then incorporated in the parsing as an alternative distribution on which the parser is trained.

- We only focused on one graph-based dependency parser throughout the thesis. However, we remarked that a transition-based dependency parser could have more difficulties with the resolution of PPA. This would need to be explored as well, and it could yield a greater impact of the PPA disambiguation technique, provided of course that the transition-based parser really performs worse on these cases.

- Finally, a natural continuation of our work is to move away from the concrete case of PP-attachment and to develop models that can cope with other types of structural ambiguity for which the parser error rate is above the average, or even, with the structural ambiguity in general.

-

Bibliography

[Abeillé & Barrier 2004] A. Abeillé and N. Barrier. *Enriching a French treebank.* In Proceedings of the International Conference on Language Resources and Evaluation, LREC 2004, pages 2233–2236, 2004.

[Agirre *et al.* 2008] Eneko Agirre, Timothy Baldwin and David Martínez. *Improving Parsing and PP Attachment Performance with Sense Information.* In Kathleen McKeown, Johanna D. Moore, Simone Teufel, James Allan and Sadaoki Furui, editeurs, ACL, pages 317–325. The Association for Computer Linguistics, 2008.

[Altmann & Steedman 1988] G. T. M. Altmann and M. Steedman. *Interaction with context during human sentence processing.* Cognition, vol. 30, no. 3, pages 191–238, 1988.

[Atterer & Schütze 2007] Michaela Atterer and Hinrich Schütze. *Prepositional phrase attachment without oracles.* Computational Linguistics, vol. 33, no. 4, pages 469–476, 2007.

[Baroni & Zamparelli 2010] Marco Baroni and Roberto Zamparelli. *Nouns are vectors, adjectives are matrices: representing adjective-noun constructions in semantic space.* In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10, pages 1183–1193, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[Bharati *et al.* 2005] Akshar Bharati, U. Rohini, P. Vishnu, Sushma Bendre and Rajeev Sangal. *A Hybrid Approach to Single and Multiple PP Attachment Using WordNet.* In Robert Dale, Kam-Fai Wong, Jian Su and Oi Yee Kwong, editeurs, IJCNLP, volume 3651 of *Lecture Notes in Computer Science*, pages 211–222. Springer, 2005.

[Bikel 2004] Daniel M. Bikel. *Intricacies of Collins' Parsing Model.* Computational Linguistics, vol. 30, no. 4, pages 479–511, 2004.

[Bohnet & Kuhn 2012] Bernd Bohnet and Jonas Kuhn. *The Best of BothWorlds - A Graph-based Completion Model for Transition-based Parsers.* In Walter Daelemans, Mirella Lapata and Lluís Màrquez, editeurs, EACL, pages 77–87. The Association for Computer Linguistics, 2012.

[Bohnet 2010] Bernd Bohnet. *Top Accuracy and Fast Dependency Parsing is not a Contradiction.* In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 89–97, Beijing, China, August 2010. Coling 2010 Organizing Committee.

[Brill & Resnik 1994] Eric Brill and Philip Resnik. *A rule-based approach to prepositional phrase attachment disambiguation.* In Proceedings of the 15th conference on Computational linguistics - Volume 2, COLING '94, pages 1198–1204, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[Bullinaria & Levy 2007] J.A. Bullinaria and J.P. Levy. *Extracting semantic representations from word co-occurrence statistics: A computational study.* Behavior Research Methods, no. 3, page 510, 2007.

[Calvo & Gelbukh 2003] Hiram Calvo and Alexander F. Gelbukh. *Improving Prepositional Phrase Attachment Disambiguation Using the Web as Corpus.* In Alberto Sanfeliu and José Ruiz-Shulcloper, editeurs, CIARP, volume 2905 of *Lecture Notes in Computer Science*, pages 604–610. Springer, 2003.

[Candito & Seddah 2010] Marie Candito and Djamé Seddah. *Parsing word clusters.* In Proceedings of the NAACL-HLT 2010 First Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010), pages 76–84, Los Angeles, United States, 2010. ACL.

[Candito *et al.* 2010] Marie Candito, Benoît Crabbé and Pascal Denis. *Statistical French Dependency Parsing: Treebank Conversion and First Results.* In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner and Daniel Tapias, editeurs, LREC. European Language Resources Association, 2010.

[Cerisara & Gardent 2009] Christophe Cerisara and Claire Gardent. *Analyse syntaxique du français parlé.* In Journée ATALA, Paris, France, 2009.

[Church & Hanks 1990] Kenneth Ward Church and Patrick Hanks. *Word association norms, mutual information, and lexicography.* Computational Linguistics, vol. 16, no. 1, pages 22–29, 1990.

[Collins & Brooks 1995] Michael Collins and James Brooks. *Prepositional phrase attachment through a backed-off model.* In Proceedings of the Third Workshop on Very Large Corpora, pages 27–38, Somerset, New York, 1995.

[Collins 1997] Michael Collins. *Three generative, lexicalised models for statistical parsing.* In Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics, EACL '97, pages 16–23, Stroudsburg, PA, USA, 1997. Association for Computational Linguistics.

[Coppola *et al.* 2011] Gregory F. Coppola, Alexandra Birch, Tejaswini Deoskar and Mark Steedman. *Simple semi-supervised learning for prepositional phrase attachment.* In Proceedings of the 12th International Conference on Parsing Technologies, IWPT '11, pages 129–139, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[Cruse 1986] D.A. Cruse. Lexical semantics. Cambridge University Press, Cambridge, UK, 1986.

[Davis & Goadrich 2006] Jesse Davis and Mark Goadrich. *The relationship between Precision-Recall and ROC curves.* In Proceedings of the 23rd international conference on Machine learning, ICML '06, pages 233–240, New York, NY, USA, 2006. ACM.

[Deerwester *et al.* 1990] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas and Richard A. Harshman. *Indexing by Latent Semantic Analysis.* Journal of the American Society for Information Science (JASIS), vol. 41, no. 6, pages 391–407, 1990.

[Dunning 1993] Ted Dunning. *Accurate methods for the statistics of surprise and coincidence.* Computational Linguistics, vol. 19, no. 1, pages 61–74, 1993.

[Erk & Padó 2008] Katrin Erk and Sebastian Padó. *A structured vector space model for word meaning in context.* In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08, pages 897–906, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[Evert & Lenci 2009] Stefan Evert and Alessandro Lenci. *Foundations of distributional semantic models.* Tutorial at ESSLLI 2009, Bordeaux, 2009.

[Evert 2005] Stefan Evert. *The statistics of word cooccurrences: Word pairs and collocations.* PhD thesis, Universität Stuttgart, Institut für Maschinelle Sprachverarbeitung, 2005.

[Fabre & Frérot 2002] C. Fabre and C. Frérot. *Groupes prépositionnels arguments ou circonstants : vers un repérage automatique en corpus.* In Actes du colloque TALN, pages 215–224, Nancy, 2002.

[Firth 1957] J.R. Firth. *A synopsis of linguistic theory 1930-1955.* Studies in linguistic analysis, pages 1–32, 1957.

[Foth & Menzel 2006] Kilian A. Foth and Wolfgang Menzel. *The Benefit of Stochastic PP Attachment to a Rule-Based Parser.* In Nicoletta Calzolari, Claire Cardie and Pierre Isabelle, editeurs, ACL. The Association for Computer Linguistics, 2006.

[Gala & Lafourcade 2007] Nuria Gala and Mathieu Lafourcade. *PP Attachment Ambiguity Resolution with Corpus-based Pattern Distributions and Lexical Signatures.* ECTI-CIT Transactions on Computer and Information Technology, vol. 2, no. 2, pages 116–120, 2007.

[Gamallo *et al.* 2003] Pablo Gamallo, Alexandre Agustini and José Gabriel Pereira Lopes. *Acquiring Semantic Classes to Elaborate Attachment Heuristics.* In Fernando Moura-Pires and Salvador Abreu, editeurs, EPIA, volume 2902 of *Lecture Notes in Computer Science*, pages 479–487. Springer, 2003.

[Harris 1954] Zellig Harris. *Distributional structure.* Word, vol. 10, no. 23, pages 146–162, 1954.

[Henestroza & Candito 2011] Enrique Henestroza and Marie Candito. *Parse Correction with Specialized Models for Difficult Attachment Types.* In EMNLP, pages 1222–1233. ACL, 2011.

[Hindle & Rooth 1993] Donald Hindle and Mats Rooth. *Structural ambiguity and lexical relations.* Computational Linguistics, vol. 19, pages 103–120, 1993.

[Hirst 1987] Graeme Hirst. Semantic interpretation and the resolution of ambiguity. Cambridge University Press, New York, NY, USA, 1987.

[Jurafsky & Martin 2008] Daniel Jurafsky and James H. Martin. Speech and language processing (2nd edition). Prentice Hall, 2 édition, 2008.

[Kawahara & Kurohashi 2005] Daisuke Kawahara and Sadao Kurohashi. *PP-Attachment Disambiguation Boosted by a Gigantic Volume of Unambiguous Examples.* In IJCNLP'05, pages 188–198, 2005.

[Kilgarriff 1997] Adam Kilgarriff. *"I Don't Believe in Word Senses".* Computers and the Humanities, vol. 31, no. 2, pages 91–113, 1997.

[Kilgarriff 2007] Adam Kilgarriff. *Googleology is Bad Science.* Computational Linguistics, vol. 33, no. 1, pages 147–151, 2007.

[Kübler *et al.* 2009] Sandra Kübler, Ryan T. McDonald and Joakim Nivre. Dependency parsing. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2009.

[Landauer *et al.* 2007] T. K. Landauer, D. S. McNamara, S. Dennis and W. Kintsch. Lawrence Erlbaum, Mahwah, New Jersey, 2007.

[Lenci 2008] Alessandro Lenci. *Distributional approaches in linguistic and cognitive research.* Italian Journal of Linguistics, vol. 20, no. 1, pages 1–31, 2008.

[Lund & Burgess 1996] Kevin Lund and Curt Burgess. *Producing high-dimensional semantic spaces from lexical co-occurrence.* Behavior Research Methods, Instruments, and Computers, vol. 28, no. 2, pages 203–208, 1996.

[Manning & Schütze 1999] Christopher D. Manning and Hinrich Schütze. Foundations of statistical natural language processing. MIT Press, Cambridge, MA, USA, 1999.

[Marcus *et al.* 1993] Mitchell P. Marcus, Mary Ann Marcinkiewicz and Beatrice Santorini. *Building a large annotated corpus of English: the penn treebank.* Computational Linguistics, vol. 19, no. 2, pages 313–330, 1993.

[McDonald *et al.* 2005] Ryan McDonald, Fernando Pereira, Kiril Ribarov and Jan Hajic. *Non-Projective Dependency Parsing using Spanning Tree Algorithms.* In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pages 523–530, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

[Medimi & Bhattacharyya 2007] Srinivas Medimi and Pushpak Bhattacharyya. *A flexible unsupervised PP-attachment method using semantic information.* In Proceedings of the 20th international joint conference on Artifical intelligence, IJCAI'07, pages 1677–1682, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[Merlo 2003] Paola Merlo. *Generalised PP-attachment disambiguation using corpus-based linguistic diagnostics.* In Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1, EACL '03, pages 251–258, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[Mitchell & Lapata 2008] Jeff Mitchell and Mirella Lapata. *Vector-based Models of Semantic Composition.* In Proceedings of ACL-08: HLT, pages 236–244, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[Mitchell & Lapata 2010] Jeff Mitchell and Mirella Lapata. *Composition in Distributional Models of Semantics.* Cognitive Science, vol. 34, no. 8, pages 1388–1429, 2010.

[Nivre *et al.* 2007] Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel and Deniz Yuret. *The CoNLL 2007 Shared Task on Dependency Parsing.* In Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, pages 915–932, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[Nivre 2005] Joakim Nivre. *Dependency Grammar and Dependency Parsing.* Rapport technique, Växjö University: School of Mathematics and Systems Engineering, 2005.

[Nivre 2006] Joakim Nivre. Inductive dependency parsing (text, speech and language technology). Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[Olteanu & Moldovan 2005] Marian Olteanu and Dan Moldovan. *PP-attachment disambiguation using large context.* In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT '05, pages 273–280, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[Padó & Lapata 2003] Sebastian Padó and Mirella Lapata. *Constructing semantic space models from parsed corpora.* In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1, ACL '03, pages 128–135, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[Padó & Lapata 2007] Sebastian Padó and Mirella Lapata. *Dependency-Based Construction of Semantic Space Models.* Computational Linguistics, vol. 33, pages 161–199, 2007.

[Pantel & Lin 2000] Patrick Pantel and Dekang Lin. *An unsupervised approach to prepositional phrase attachment using contextually similar words.* In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, ACL '00, pages 101–108, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

[Ratnaparkhi *et al.* 1994] Adwait Ratnaparkhi, Jeff Reynar and Salim Roukos. *A maximum entropy model for prepositional phrase attachment.* In Proceedings of the workshop on Human Language Technology, HLT '94, pages 250–255, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[Ratnaparkhi 1998] Adwait Ratnaparkhi. *Statistical models for unsupervised prepositional phrase attachment.* In Proceedings of the 17th international conference on Computational linguistics - Volume 2, COLING '98, pages 1079–1085, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

[Sahlgren 2006] Magnus Sahlgren. *The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-Dimensional Vector Spaces.* PhD thesis, Stockholm University, Stockholm, Sweden, 2006.

[Schmid 1994] Helmut Schmid. *Probabilistic Part-of-Speech Tagging Using Decision Trees.* In Proceedings of the International Conference on New Methods in Language Processing, pages 44–49, 1994.

[Schütze 1993] Hinrich Schütze. *Word space.* In Advances in Neural Information Processing Systems 5, 1993.

[Schütze 1995] Carson T. Schütze. *PP attachment and argumenthood.* MIT Working Papers in Linguistics, vol. 26, pages 95–151, 1995.

[Stetina & Nagao 1997] Jiri Stetina and Makoto Nagao. *Corpus Based PP Attachment Ambiguity Resolution with a Semantic Dictionary.* In Joe Zhou and Kenneth W. Church, editeurs, Proceedings of the Fifth Workshop on Very Large Corpora, pages 66–80, Beijing, China, 1997. ACL.

[Søgaard 2011] Anders Søgaard. *Using graphical models for PP attachment.* In The 18th Nordic Conference on Computational Linguistics, pages 206–213, 2011.

[Tesnière 1959] Lucien Tesnière. Elements de syntaxe structurale. Editions Klincksieck, 1959.

[Toutanova *et al.* 2004] Kristina Toutanova, Christopher D. Manning and Andrew Y. Ng. *Learning random walk models for inducing word dependency distributions.* In Proceedings of the twenty-first international conference on Machine learning, ICML '04, pages 815–822, New York, NY, USA, 2004. ACM.

[Toutanova 2006] Kristina Toutanova. *Competitive generative models with structure learning for NLP classification tasks.* In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP '06, pages 576–584, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[Turney & Pantel 2010] Peter D. Turney and Patrick Pantel. *From frequency to meaning: vector space models of semantics.* Journal of Artificial Intelligence Research, vol. 37, pages 141–188, 2010.

[Van de Cruys 2009] Tim Van de Cruys. *A non-negative tensor factorization model for selectional preference induction.* In Proceedings of the Workshop on Geometrical Models of Natural Language Semantics, GEMS '09, pages 83–90, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[van Herwijnen *et al.* 2003] Olga van Herwijnen, Jacques Terken, Antal van den Bosch and Erwin Marsi. *Learning PP attachment for filtering prosodic phrasing.* In Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1, EACL '03, pages 139–146, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[Volk 2002] Martin Volk. *Combining unsupervised and supervised methods for PP attachment disambiguation.* In Proceedings of the 19th international conference on Computational linguistics - Volume 1, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[Widdows 2004] Dominic Widdows. Geometry and meaning. Center for the Study of Language and Information, Stanford, CA, 2004.

[Zeldes 2009] Amir Zeldes. *Quantifying constructional productivity with unseen slot members.* In Proceedings of the Workshop on Computational Approaches to Linguistic Creativity, CALC '09, pages 47–54, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[Zhao & Lin 2004] Shaojun Zhao and Dekang Lin. *A Nearest-Neighbor Method for Resolving PP-Attachment Ambiguity.* In IJCNLP, pages 545–554, 2004.

[Ângelo Mendonça *et al.* 2009] Ângelo Mendonça, David Graff and Denise DiPersio. *French Gigaword Second Edition*, 2009.