TEL AVIV UNIVERSITY אוניברסיטת תל-אביב
The Raymond and Beverly Sackler Faculty of Exact Sciences
The Blavatnik School of Computer Science

Prepositional-Phrase Attachment Disambiguation Using Derived Semantic Information and Large External Corpora

This thesis is submitted in partial fulfillment of the requirements for the M.Sc. degree in the School of Computer Science, Tel Aviv University

by

Lena Dankin

This research for this thesis has been carried out at Tel Aviv University under the supervision of Prof. Nachum Dershowitz

June 2015

Contents

1. Introduction

Prepositional phrase (henceforth PP) attachment disambiguation is an important task within the task of syntactic parsing of text. Based on the British National Corpus (BNC) [9], out of the top-ten most frequent words in English, four are prepositions (of, to, in and for). The frequency of the prepositions in the text emphasizes the need for correct PP attachment during the parsing process, since it affects the resulted parse tree. An incorrect attachment can have a major influence in several linguistic tasks that embed syntactic parsing, such as information retrieval.

Clearly, PP attachment disambiguation is not the only challenge in syntactic parsing. However, they still fail to accomplish a correct disambiguation, in comparison to the accuracy of the construction of the other parts in the parse tree [15].

The problem of attachment ambiguity occurs when the syntactic rules allow more than one possible attachment for a single PP. Although each PP can have several attachment candidates, most of the PP attachment research has focused on the case of a single PP occurring immediately after an noun phrase, which in turn is preceded by a verb (thus the candidates are either the verb or the noun). Such an approach requires an oracle that provides the two hypothesized structures (noun and verb) that we choose between. These candidates are usually extracted from the gold standard parse trees, or detected by the parser as it tries to apply the PP attachment algorithm to attach or reattach a PP during the parsing process. When the parser fails to detect such a tuple, the disambiguation process will not be executed, which weakens the algorithm. Nevertheless, this binary definition of the problem covers most of the cases when syntactic parsers fail to attach the PP correctly. The oracle hypothesis is easy to detect, so we will also focus on this definition of the PP attachment problem. This task can be viewed as the binary classification of the PP attachment - noun or verb. The success rate on the oracle-free versions is lower than on the binary ones [3].

In order to understand the difficulty of the PP attachment resolution, it is useful to consider two examples (from [21]) “I ate a pizza with anchovies” vs. “I ate a pizza with friends”. These two sentences are of the same structure in terms of POS tags. Nonetheless, the trees are different (Figure 1.1) and this is only due to the semantic difference between the nouns “friends” and “anchovies”. Anchovies are a common pizza

topping. Therefore, “with anchovies” is attached to the word “pizza” rather than “ate”. As for the noun “friends”, we do not usually consider friends as an eatable object, so the attachment will be to the verb. This difference is very clear to us, humans, due to our vast common knowledge. However, naturally a syntactic parser will have to disambiguate these cases using semantic techniques. These two sentences demonstrate the fact that syntactic parsing is not purely a syntactic task, and it can also benefit from semantic information.

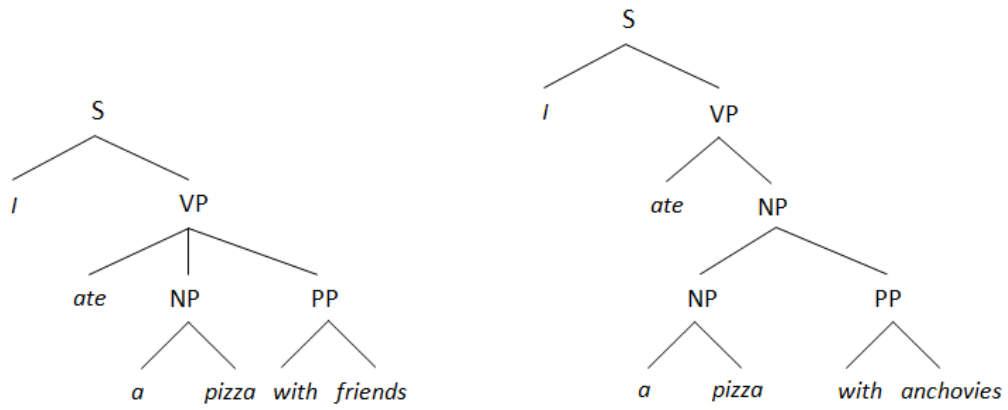

Figure 1.1: Parse trees for "I ate a pizza with friends" and "I ate a pizza with anchovies". While the part-of-speech tags are identical, we get two different parse trees.

An even more complicated problem unfolds in the following two examples: “I saw a girl with a suitcase” vs. “I saw a girl with a telescope”. While we all know that the one who is holding the suitcase is the girl, the telescope possession cannot be determined based on this single sentence. In the second sentence, however, both attachments will be correct in terms of the syntax and the semantics, and only the context of this sentence might provide the required information to the ambiguity resolution. An example for such context will be a sentence in the same paragraphs that mentions who is in possession of the telescope. Despite the fact that the usual input for a parser is only the sentence, we claim that, whenever possible, having the context may improve the results, for instance, as a part of a discourse parsing or an analysis of a complete article.

The methods that are commonly used to resolve the disambiguation are rule-based and statistical classification algorithms, comprised of supervised, semi-supervised and unsupervised learning (see Chapter 2). Naturally, we aim to learn as much as possible from an unlabeled corpora, since such corpora are incomparably larger than the labeled ones.

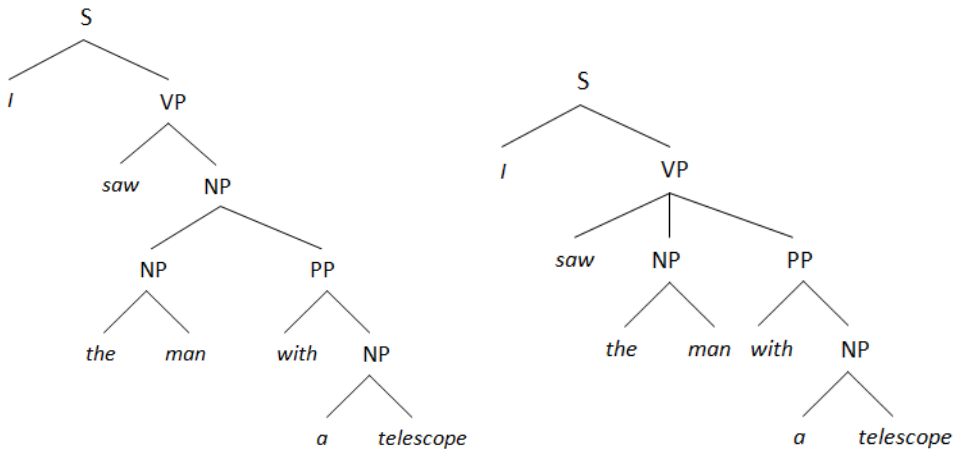

Figure 1.2: Two possible (and correct) parse trees for "I saw the man with a telescope"

We present our method to derive such data from the British National Corpus, relying on previously introduced approaches but with substantial modifications. Despite the lack of the correct parse tree for the BNC sentences, we introduce a technique that uses them in addition to, or instead of, the labeled training corpus, which is rather small. We use the BNC to extract two sets of examples: ambiguous and unambiguous, classified according to an algorithm presented in this paper. On the ambiguous examples, we estimate the distribution of both classes (noun and verb attachment) using a syntactic parser. This is an estimation because the parser is not accurate (or else all work on PP attachment would be redundant), yet, since the parser attaches the PP correctly is most cases, which is sufficient for us. In addition, we introduce a method to embed the sentence's context, when available, in an attempt to derive more knowledge regarding each prepositional phrase, and consequently to increase the accuracy.

Our results are reported on two datasets. One is the standard benchmark for the binary definition of the pp attachment disambiguation problem, and the other is a dataset that was constructed by us from the WSJ tagged corpus. The standard dataset (developed by Ratnaparkhi et al. [24]) contains quadruplets of the form <v, n1, p, n2> along with the correct attachment (v and n1 are the attachment candidates, p is the preposition and n2 is the head noun of the PP). The additional dataset was constructed mainly because the alignment to the original sentences is unavailable and cannot be done perfectly due to the many changes in the corpus over the years.

The structure of this work is as follows: Chapter 2 presents previous work on the pp attachment problem and prior results (we usually measure accuracy). Chapter 3 describes all of the linguistic tools that we use and the algorithms we apply. The results are summarized in Chapter 4. Chapter 5 contains conclusions and ideas for future work.

2. Literature Review

In this section we review several relevant works on the task of PP-disambiguation.

Hindle and Rooth (1993) suggested that many ambiguous prepositional phrase attachments can be resolved on the basis of the relative strength of association of the preposition with verbal and nominal heads, estimated on the basis of distribution in an automatically parsed corpus [14]. Learning from a large corpus (AP, 13 million words), they look at the log of the ratio of the probability of verb attach to the probability of noun attach, based on the candidates and the preposition. The PP will be attached to the more probable candidate.

In order to overcome the problem of sparse data, an interpolation is being used - the probability of a single candidate attached to the preposition is interpolated with the probability of a noun/verb attachment for this preposition, covering the cases when the candidate is absent from the corpus, or the candidate is never attached to the preposition.

Hindle and Rooth's method required no explicitly annotated training data, nor did it use any semantic or syntactic processing of the text. Moreover, they didn't use the noun object of the preposition (n2) at all in the disambiguation process. Their reported accuracy on the RRR set is 79.7.

Ratnaparkhi, Reynar and Roukos (1994) introduced a Maximum Entropy (ME) model which attempted to predict the probability of attachment decisions by constructing statistical models [24]. Their model only made use of the lexical information within verb phrases, and did not depend on any external semantic knowledge base. They extracted verb-phrases with PP-attachment ambiguities from the Penn Treebank WSJ corpus and the IBM-Lancaster Treebank including the attachment information, and constructed the test and training datasets. The dataset, which consisted of 27,937 quadruplets, has since established itself as a benchmark dataset. The model which was based on models of exponential family constructed using the Maximum Entropy Principle, then assigned a probability to either of the possible attachments. The ME model produces a probability distribution for the PP-attachment decision using only the information from the ambiguous verb phrases in question. The experiment produced satisfactory results with the ME model predicting PP-attachments with 78.0% accuracy, compared to an average lexicographer performing at 88.2% accuracy. For comparison, they obtained PP-

attachment resolution performances of three TreeBank experts on a set of three hundred randomly selected test events from the WSJ corpus.

A test conducted with the human experts provided the following lower bounds of the performance on the data: always choosing noun attachment yields the precision of 59%. Choosing the most likely attachment for each preposition yields the precision of 72.2%. Another two interesting bounds were achieved while checking human attachment accuracy on a subset of the RRR corpus. Humans reached the accuracy of 88.2% provided only the quadruplets, and 93.2% provided both the quadruplets and the sentence from WSJ the quadruplet was extracted from.

Nakov and Hearst (2005) proposed a method to resolve PP-attachment ambiguity using unsupervised algorithms which exploited the WWW as a very large corpus, making use of its surface features and paraphrases[20]. This was based on the assumption that phrases found on the WWW are sometimes disambiguated and annotated by content creators. Their experiment used n-gram models, where statistics were obtained by querying exact phrases including inflections and all possible variations derived from WordNet, against WWW search engines, using WordNet to extract synonyms from word sense hierarchies and to construct all possible variations of a given phrase. The accuracy of their statistical algorithm produced an average accuracy of 83.82% on the RRR data set.

Collins and Brooks (1995) introduce the back-off model as a statistical approach for a PP attachment problem represented by a quadruplet of 4 head words - (v, n1, p, n2), the same as [21]. They suggested that the problem is analogous to n-gram language models in speech recognition, and that one of the most common methods for language modeling, the backed-off estimate, is applicable. Backed-off n-gram word models for speech recognition are used to estimate the probability of the next word in a text given the (n-l) preceding words. This will enable using MLE on a sparse data, backing off to smaller n-grams if the counts are not high enough to make an accurate estimate at the current level. In a similar manner, they calculate the probability of each attachment for the tuple by starting with triplets counts, and backing off to pairs in case the triplets were not found in the corpus.

The overall accuracy of their method is 84.1%. When analyzing the results according to the tuples that were detected in the training corpus, it is evident that the larger the detected tuple is, the higher its accuracy.

Brill and Resnik (1995) presented a rule based corpus based approach to disambiguate a PP attachment [7]. The patterns that are used as rules are being learned using a

transformation- based error driven model. At the first stage, all PP are attached to the noun. Next, a set of transition patterns are being learned and scores, based on the error rate. Each pattern corresponds to a possible transition (from noun to verb attachment and vice versa). All patterns are generated from pre-defined templates. An example of a pattern template is "change attachment from X to Y if v is W" and the learned pattern is "change attachment from n1 to v if v is put". In order to overcome the data sparseness in the training process, they added class information for nouns, taken from WordNet. Each noun was represented by a set of its hypernyms and the pattern was also extended to matching items in the hypernyms set (i.e. "v is a part of C"). The match had a boolean value, meaning that a word can either be contained in the hypernyms set or not. More delicate similarity measures were not tried.

Stetina and Nagao (1997) propose a supervised learning method for PP attachment based on a semantically tagged corpus [25]. Their idea was to improve the performance of the back-off model developed by Collins and Brooks by increasing the percentage of full quadruplet and triple matches by employing a semantic distance measure. As a part of the algorithm, a sense disambiguation procedure was executed on the tuple words, in order to improve the accuracy of the query expansion. The sense disambiguation was performed using contextual similarity between ambiguous words (ambiguity is defined by multiple senses in WordNet). The pp disambiguation itself was performed using decision trees, and the reported accuracy is 88.1%. As they state in the paper, this accuracy is partly attributed to the positive bias of disambiguation of the testing examples against the same training set which is also used for the decision tree induction.

The disambiguation errors are thus hidden by their replication in both the training and the testing sets.

Ratnaparkhi (1998) proposed an unsupervised approach that uses a heuristic based on attachment proximity and trains from raw text annotated with only part-of-speech tags and morphological base forms, as opposed to attachment information [23]. After POS tagging and chunking the raw corpus, they count all <candidate, p, n2> triplets. As opposed to Stetina and Nagao, they only use unambiguous counts in the raw corpus, based on the hypothesis that the information in just the unambiguous attachment events can resolve the ambiguous attachment events of the test data. The accuracy reported on the RRR data set is 83.7%.

Pantel and Lin (2000) presented an unsupervised corpus-based approach to pp attachment using an iterative process to extract training data from an automatically parsed corpus [22]. They use a collocation database to determine contextually similar

words to the noun and verbs in each quadruplet. In addition, using a large corpus, two datasets of the counts of triples of the form (candidate, p, n2) are created. The first one counts ambiguous cases (where the candidate appears within a short distance from the pp, but it may not be the correct attachment). The second only counts unambiguous cases.

The attachment decision for a 4-tuple (v, n1, p, n2) is made in two stepa. First, v and n2 are replaced by their contextually similar words and compute the average adverbial attachment score. Similarly, the average adjectival attachment score is computed by replacing n1 and n2 by their contextually similar words. The attachment is determined by the combination of average scores for each attachment candidate. The candidate with the higher score is selected. The accuracy of this method is 84.31% on the RRR dataset.

Olteanu and Moldovan (2005) introducea new approach to disambiguate a pp attachment using a Support Vector Machine learning model that uses complex syntactic and semantic features as well as unsupervised information obtained from the World Wide Web [21]. Results were provided for three datasets - the benchmark RRR data set, a data set extracted from WSJ, aligned with the original sentence, and a data set extracted from FrameNet. Each data set enabled the usage of additional features - in the data set extracted from FrameNet, they used the semantic frames of each sentence in order to capture the semantic behavior of the verb candidate [5]. As for the dataset extracted from WSJ, they were able to use data extracted from the gold standard parse tree, but without using the actual information about the correct attachment. The queries to the WWW were generated from the quadruplet input, using lemmatization, without any semantic expansions. They report the accuracy of 92.83% and % 93.62 on both datasets that were have created for this task, comparing the results with an accuracy of 86.1%, implementing Collins and Brooks back-off method on each of them.

Zhao and Lin (2004) presented a nearest neighbor algorithm for resolving pp attachment ambiguity, using the cosine of the pointwise mutual information vector that represents each word (as well as with other common similarity measures, that performed worse than the cosine similarity) [27]. Given a quadruplet to be classified, they search the training classified examples to its top-k nearest neighbors and determine its attachment base of the known classifications of the nearest neighbors. The similarity between two quadruples is determined by the distributional similarity between the corresponding words in the quadruples. The reported accuracy of their method is 86.5% on the RRR data set.

Bharathi et al. (2005) proposed an algorithm that uses a combination of supervised and unsupervised learning, along with information from WordNet [6]. Given a quadruplet, they check if it exists in the supervised data, and use that assigned tag. Otherwise, a back-off model is employed using the probabilities of smaller sub-queries in the labeled corpus, as well as searched in a large unlabeled corpus (the unsupervised stage). Their accuracy is 84.6% on the RRR dataset, and 86.44% and 88.99% on two additional data sets they extracted for this task.

3. Methods

3.1 The corpora

WSJ corpus

The WSJ corpus is a collection of Wall Street Journal articles from 1987 to 1989 [10]. Out of 98,732 stories, a total of 2,499 were selected for syntactic annotation (see Section 3.1.2.1), and this subset or stories is widely used in various NLP tasks, among them also PP disambiguation.

Penn Treebank:

Annotated Corpus

The Penn TreeBank is a large annotated corpus, constructed between 1989 and 1996 from a wide range of texts, such as the *Wall Street Journal* articles, IBM computer manuals, transcribed telephone conversations, etc. There are nearly 7M words of text annotated for POS, and 3M words of text syntactically annotated.
The annotation process of this corpus was a two stage process for both POS and semantic tagging. First, an automated tagging was executed; afterwards a manual correction was performed by linguists.

Annotations

We distinguish between two sorts of annotations: POS (part-of-speech) annotation and syntactic annotations.
POS tagging is the process of marking each word of the sentence with its part-of-speech based on its context in the sentence. The Penn Treebank tag set is considered to be one of the most popular tag sets for English. It consists of 36 POS tags and 12 other tags for punctuation and currency symbols [18] (see Figure 3.1).
Syntactic parsing (also referred to as “syntactic bracketing”) is the process of the analysis of a sentence according to a predefined formal grammar. In this task, phrases are being tagged according to a predefined syntactic tag set. Penn Treebank provides a set of 17 syntactic tags (see Figure 3.2).

CC	Coordinating conj.	TO	infinitival *to*
CD	Cardinal number	UH	Interjection
DT	Determiner	VB	Verb, base form
EX	Existential there	VBD	Verb, past tense
FW	Foreign word	VBG	Verb, gerund/present pple
IN	Preposition	VBN	Verb, past participle
JJ	Adjective	VBP	Verb, non-3rd ps. sg. present
JJR	Adjective, comparative	VBZ	Verb, 3rd ps. sg. present
JJS	Adjective, superlative	WDT	Wh-determiner
LS	List item marker	WP	Wh-pronoun
MD	Modal	WP$	Possessive *wh*-pronoun
NN	Noun, singular or mass	WRB	Wh-adverb
NNS	Noun, plural	#	Pound sign
NNP	Proper noun, singular	$	Dollar sign
NNPS	Proper noun, plural	.	Sentence-final punctuation
PDT	Predeterminer	,	Comma
POS	Possessive ending	:	Colon, semi-colon
PRP	Personal pronoun	(	Left bracket character
PP$	Possessive pronoun	)	Right bracket character
RB	Adverb	"	Straight double quote
RBR	Adverb, comparative	‘	Left open single quote
RBS	Adverb, superlative	“	Left open double quote
RP	Particle	’	Right close single quote
SYM	Symbol	”	Right close double quote

Figure 3.1: pos tags annotation from Pen TreeBank

ADJP	Adjective phrase
ADVP	Adverb phrase
NP	Noun phrase
PP	Prepositional phrase
S	Simple declarative clause
SBAR	Subordinate clause
SBARQ	Direct question introduced by *wh*-element
SINV	Declarative sentence with subject-aux inversion
SQ	Yes/no questions and subconstituent of SBARQ excluding *wh*-element
VP	Verb phrase
WHADVP	Wh-adverb phrase
WHNP	Wh-noun phrase
WHPP	Wh-prepositional phrase
X	Constituent of unknown or uncertain category
*	“Understood” subject of infinitive or imperative
0	Zero variant of *that* in subordinate clauses
T	Trace of wh-Constituent

Figure 3.2: syntactic tags from Pen TreeBank

RRR Data set

The most popular and widely used benchmark for the PP attachment disambiguation problem was created by Ratnaparkhi, Reynar and Roukos [24]. This corpus contains quadruplets of the form <V, N1, P, N2>, where V and N1 are the attachment candidates and P and N2 are the preposition and head noun from the prepositional phrase

respectively. The tuples were automatically extracted from the Penn Treebank Wall Street Journal (WSJ) corpus, and each tuple appears with the correct attachment type – noun or verb, derived from the manual annotations of the WSJ corpus. The quadruplets dataset contains 20801 tuples for training, 4039 tuples for development and 3097 tuples for testing.

It is known that the RRR data contains some noise, yet no corrected benchmark was published (Ratnaparkhi 1998, Stetina and Nagao, 1997). The noise can be divided into two categories: wrong extraction of the tuple, and wrong classification of the attachment. In the test corpus, we can find the word "the" appears 161 times as one of the nouns (n1 or n2). In addition, we can find "'s" 30 times as a noun.

As for a possible misclassification, the quadruplet <requested, treatment, for, types> was extracted from the sentence "Timex had requested duty-free treatment for many types of watches, covered by 58 different U.S. tariff classifications". While the gold standard attaches the pp to the noun (see Figure 3.3), in the RRR dataset, the quadruplet is classified as a verb attachment.

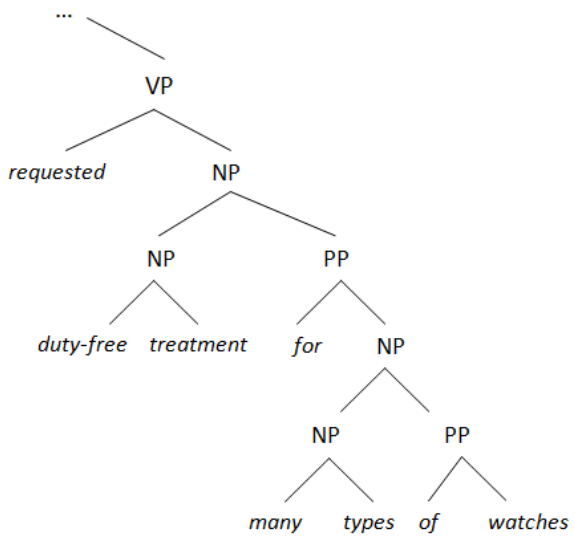

Figure 3.3: a part of the gold standard parse tree to the sentence "Timex had requested duty-free treatment for many types of watches, covered by 58 different U.S. tariff classifications", focused on the attachment of "for many types of watches". The pp is a part of a noun phrase and is attached to "duty free treatment".

WSJ-sent Dataset

In order to overcome problems within the RRR dataset, as well as to align the quadruplets with their original sentence, we've extracted a new dataset from the WSJ (the same as Olteanu and Moldovan did; their data set was unavailable to us). It will be further referred to as the WSJ-sent dataset.

The data was extracted by the following method: using the gold standard parse trees from the WSJ corpus, we extract all prepositional phrases from all sentences. For each pp, we first detect the correct attachment, examining the governor of the pp, that is, the noun or the verb that the pp is attached to. We detect the governor by acceding from the pp node in the gold parse tree, until a node tagged as a noun phrase or a verb phrase is reached.

Next, we check for a possible ambiguity, since we are only interested in pp attachment cases with more than one possible attachment candidate, one being a noun and one being a verb. For a verb governor, we search for a noun phrase that is also governed by the same verb, and is located between the verb and the pp. For a noun governor, we search for its verb governor (in cases like "the reservation for four diners was made two hours ago", there is no ambiguity in the attachment, since "for four" can only be attached to the noun "reservation").

Overall, we've extracted 61,601 tuples, and divided them uniformly into three sets: train (80%), development (10%) and test (10%). Each tuple is aligned with its original sentence, and we also keep track of its context (a window of adjacent sentences). The original sentence, as well the as context, will be used by some of our features.

BNC

The BNC (British National Corpus) is a 100-million-word collection of samples of written and spoken language from a wide range of sources, designed to represent a wide cross-section of both spoken and written British English from the latter part of the 20th century[9]. The BNC is provided with POS annotation and the lemma (lemma is the canonical, or dictionary form for each word) for each word.

This corpus is used in our work to build the probability distributions for each tuple, for each possible attachment. These distributions will be used as features for our classifier.

3.2 Linguistics tools and Methods

Stanford Parser

The Stanford Parser was developed by the Stanford Natural Language Processing Group [15] is considered to be reliable and is widely used. It is based on probabilistic context-free grammars (PCFGs) and provides both POS annotation and syntactic parse tree with its score. A significant advantage of the parser is its ability to return k best parse trees,

along with their score, enabling us to examine not only the selected option, but other candidates as well.

WordNet

WordNet is a large lexical database for the English language, that is still being developed at Princeton University. It contains nouns, verbs, adjectives and adverbs that are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept, all (as of now) manually annotated. as of now, It currently consists of 155,000 words.
The main relation among words in WordNet is synonymy. Synonyms are two words that denote the same concept and are interchangeable in many contexts and are grouped into unordered sets (synsets). WordNet contains a total of 117,000 different synsets.
WordNet contains more relations, such as hyponymy and hypernymy. A word is a hyponym of another word if it is its more specific term. A hypernym would be a more general term.
A word sense hierarchy is a lexical tree formed by a sequence of hypernyms in different levels, with each level being trailed by the synset of a superordinate term (hypernym) to a limited depth. That is, the last level being the most generic superordinate term of the term in the first level (see Figure 3.3)

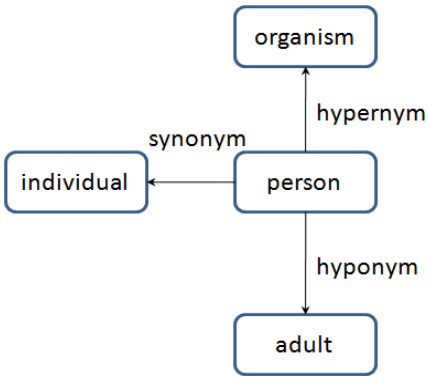

Figure 3.3

FrameNet

The FrameNet project is building a lexical database of English that is both human- and machine-readable, based on annotating examples of how words are used in actual texts[5]. It contains more than 10,000 word senses, most of them with annotated examples that show the meaning and usage, as well as the more than 170,000 manually annotated sentences that provide a unique training dataset for semantic role labeling. Although having semantic frames proves to be helpful in the attachment disambiguation, the alignment of the frame to the sentence is a rather non-trivial task that is still a challenge in NLP. (In [11] they extracted quadruplets from the FrameNet database

sentences, having the frame manually annotated by the FrameNet annotator.) We, therefore, only use the realization data from FrameNet, as described in Section 3.4.1.2.

Word Similarity

Word similarity measures can be roughly divided into two categories: semantic thesauri based methods and distributional algorithms based on the observation that semantically related words tend to appear within the same context. Context can be defined as a entire document, the paragraph or a window of some predetermined size.

Using a thesaurus like WordNet, we can calculate the distance between the two words in the parse tree[8]. However, due to the unbalancedness of the WordNet hierarchy trees, two pairs can be within different distances, but related in the same manner. Embedding the distance of the lowest common ancestor of both words from the root, or its height can improve results, but it will not completely solve the problem.

The distributional method uses a vector representation of each word. The vector can represent the documents where each word appears or the context of each instance. The context is a bag-of-words of the words within a window that contains the word. We aim to represent each word with the context of content words – words that are not frequent in the language and therefore their presence in the context window is indicative. Setting the window size presents us with a tradeoff – using a small window will certainly lead to a loss of a valuable semantic knowledge on the word, while a big window generates more noise in the representation. We choose the first non-stop word or each side of each word w as the context, as in [27].

We use the cosine similarity of Positive Pointwise Mutual Information (PPMI) to calculate the relatedness between words, a standard technique in NLP.

Cosine similarity between two vectors is calculated as follows:

$$CosSim(u,v) = \frac{\sum_{i=1}^{n} u_i \cdot v_i}{\sqrt{\sum_{i=1}^{n} u_i^2} \cdot \sqrt{\sum_{i=1}^{n} v_i^2}}$$

PMI measures the strength of the association between two words as follows:

$$pmi(i,j) = \log\left(\frac{p(i,j)}{p_{i*}, p_{*j}}\right) \quad p_{ij} = \frac{f_{ij}}{\sum_{i=1}^{R}\sum_{j=1}^{C} f_{ij}} \quad p_{i*} = \frac{\sum_{j=1}^{C} f_{ij}}{\sum_{i=1}^{R}\sum_{j=1}^{C} f_{ij}} \quad p_{*j} = \frac{\sum_{j=1}^{W} f_{ij}}{\sum_{i=1}^{R}\sum_{j=1}^{C} f_{ij}}$$

f_{ij} is the number of times the word w_i appears within the context c_i

$$ppmi(i,j) = \max(pmi(i,j), 0)$$

3.3 Data Preprocessing

As previously mentioned, we lemmatize each word in the corpora (both BNC corpus, RRR and WSJ-sent). In addition, we replace all named entities with special strings, denoting each entity type with a different string. Converting all named entities of the same type to a single string can generate more accurate statistics, since we don't need to distinguish between different numbers, names, etc.

When we have the original sentence for the quadruplet (in the WSJ-sent corpus), we can a Named Entity Recognition (NER) tool. We use Stanford NER [12] to extract entities of 7 categories: PERSON, ORGANIZATION, LOCATION, DATE, TIME, MONEY and NUMBER. Finally, all the words are converted to lower case.

For the RRR dataset (where the sentence is unavailable and thus using NER tool is not straight forward), we apply a preprocessing step, replacing all numbers with the token "YEAR" to 4-digit numbers, or "NUMBER" to the rest. In addition, all capitalized nouns are replaced with the token "NAME".

3.4 Features

The problem of PP attachment resolution, as already defined, can also be viewed as a classification problem - given a quadruplet and the sentence, we need to classify it as either a noun or verb attachment. We will use supervised machine learning to handle this problem.

Our features can be roughly divided into three categories:

1. Features that are only related to the quadruplet.
2. Feature that are related to the quadruplet and the sentence it was extracted from.
3. Features that are related to the quadruplet, sentence, and the sentence's context (meaning, the adjacent sentences in the text).

Since in the RRR corpus, only the quadruplet is available, we will only use features of type 1, while working with WSJ-sent enables us to use all three types.

3.4.1 Quadruplet Features

3.4.1.1 BNC based features

In order to determine the correct attachment for each quadruplet, it is necessary to extract the probabilities of such an attachment from the BNC. This corpus does not contain gold standard parses, nonetheless, it can still provide valuable information [1,3].

From our original quadruplet, we extract the following tuples, which will be searched in the BNC: attachment candidate + p + n2, attachment candidate + p and attachment candidate + n2. The term attachment candidate represents either n1 or v.

The motivation behind searching these sequences in the BNC is that the correct attachment is more likely to appear in the large corpus than the wrong attachment. This is only correct for attachments that can be resolved within the sentence context, which is true for most of the cases.

Query list generation

For each tuple, we create a list of queries of interest. Since we are using lemmatization, we are not concerned with verb conjugations and single/plural for nouns. We wish, however, to expand our queries with semantically related nouns and verbs. For instance, in the sentences "I ate a salad with a spoon" and "I ate a salad with a fork", the words "fork" and "spoon" are related, though not synonyms. For that purpose, we use semantic relatedness measures (see Section 3.2.4) to determine the related words of each noun and verb in the quadruplet.

We used WordNet to extract all semantically related nouns and verbs, focusing on two semantic relations types:

1. Synonyms – using each synset of the target word. A synset is a set of words that is interchangeable in some contexts without changing the truth value of the preposition in which they are embedded.
2. Hypernyms – for each synset of the word, we extract all hypernyms and the hyponyms of each such hypernym. For example: one of the senses of "fork" is described by the gloss: "cutlery used for serving and eating food". Its hypernym is "eating utensil" ("tableware implements for cutting and eating food") and this synset's hyponyms are "fork", "spoon" and "table knife". We, therefore, can refer to "spoon" as the "cousin" of "fork", since they share the same hypernym.

In either expansion type, we are using all hyponym trees for each synset, since once a synset is selected as relevant, all hyponyms are also relevant.

Choosing the correct synsets is a key point in the expansion process. Each word has several senses, some of which are totally distinct (such as for the word "bank": among its senses we can find both "sloping land" and "financial institution"). Using too many senses will result in numerous queries, which may affect the performance. Most importantly, the count for tuples will be contaminated with the counts of irrelevant search results. On the other hand, we do wish to expand our queries as much as possible, since even though we

are given a big corpus such as the BNC corpus, we are not able to find examples of each correct tuple (that is, a correct attachment + p + n2).

Choosing the correct synset involved some word sense disambiguation. Having only the quadruplet may not be enough (since we need the context of the word to decide between its senses). Using the sentence (as we do in the quadruplet + sentence feature), provided us with the needed context. When the sentence is available us, we use Lesk's algorithm [17]. The algorithm disambiguates words using the overlap rate between the gloss and the sentence.

We compile the final expansions list of each word using:

- Senses ranking (when possible): using the Lesk disambiguation algorithm, we choose top k (k = 5 synsets for each word) senses and expand them to their hyponyms.
- Related candidates ranking: for each synonym/related word, we calculate the cosine similarity between the mutual information vector representation of the original word and the related candidate. As in the senses selection, we take top k related words (k = 10).

Searching queries in the BNC corpus

In order to acquire the counts of the tuples of interest in the BNC corpus, we perform preprocessing in order to align it with the tuples input format, but also to improve performance.

The preprocessing steps are:

1. Preprocess each file within the BNC corpus using the method described in Section 3.3.
2. Index each file - Since we are dealing with multiple accesses to a big corpus (~100 million words), performing searches at runtime is not feasible performance-wise.

Next, we search for all the constructed tuples, in an attempt to capture cases were a "hit" means a high probability to attach the pp to the attachment candidate. A simple search consisting of finding each word of the query, will not be enough, even if we maintain the order. This is because the pp can be unconnected to the attachment candidate, even though they are both present in the same sentence.

We divide our search task into two tasks, as in [6]: The first task is to find the tuple within an unambiguous attachment case. For instance, consider the sentence "I traveled

by bus". In this sentence, the tuple (traveled, by, bus) represents an unambiguous attachment of the pp "by bus" to the verb "traveled", since there are no other candidates. The second task is to find the same tuple within an ambiguous case. An example could be the same tuple within the sentence "I travel to work by bus". In this case, although the correct attachment is to the verb, and thus this example can be counted as a proper example in favor of the adverbial attachment, we count this instance of the tuple as an ambiguous case.

For each sentence in the BNC corpus, and each tuple in our query list, we check the following criteria:

1. Each word from the tuple should be presented in the sentence, in the exact same order.
2. No preposition should appear in the sub-sentence induced by the tuple (starting in the first tuple word and ends with the last word), other than the preposition from the tuple.
3. If the tuple contains n2, we require no verbs between p and n2, otherwise n2 may not be the head noun of the preposition phrase that starts with p.

Implementation wise, we use an indexing system to select the relevant tuples for each sentence, rather than searching for an instance of each tuple in each sentence.

Next, we classify the instance as an ambiguous/unambiguous instance according the following set of rules (slightly modified from [10]):

The ambiguous case:

- n1_p: a verb before n1, within 5 words, no noun or verb between n1 and p
- v_p: a noun between v and p, no verb
- n1_p_n2 and v_p_n2: the mix of n1_p and p_n2, or v_p and p_n2, respectively.

The unambiguous case:

- n1_p: no verb before n1 within 5 words, no noun or verb between n1 and p
- v_p: no noun between v and p, no verb
- n1_p_n2 and v_p_n2: the mix of n1_p and p_n2, or v_p and p_n2, respectively.

Features based on unambiguous search

Based on the counts, we can calculate the probabilities of each attachment for the quadruplet.

count-ratio (from [6]): the count ratio defined as: $\log_{10}\frac{f_v}{f_n}$, f defined as:

$$f_v = \frac{c_{v,p,n2}}{c_v \cdot c_{p,n2}}, \quad f_n = \frac{c_{n,p,n2}}{c_n \cdot c_{p,n2}}$$

c represents the counts of each word/tuple. Unlike [6], we use semantic expansions, so c counts also all the instances of the expanded queries.. We still wish to distinguish between the tuples that contain the original quadruplet word and the extended tuple; thus we give more weight to the counts of the original one. Formally,

$c_x = c_x + \lambda \sum_{i \in e(x)} c_i$, e(x) = all the expansions of x, c_x - unambiguous count only (for single words we use the regular instance count in the corpus).

We set λ arbitrarily to 0.7.

%hits / dataset	no_exp, true_at	exp, true_at	no_exp, false_at	exp, false_at
RRR	219/1285	759/1591	49/622	247/890
WSJ-sent	1094/4174	1785/4431	446/2626	1519/3786

Table 3.1: Hit rates in the BNC corpus for the test sets of both datasets. exp and no_exp stand for counts with/without expansions. true_at is a notation for tuples that represent the correct attachment (for a quadruplet that should be classified as adverbial attachment, the tuples are <v,p>, <v,p,n2>). false_at stands for the opposite – tuples that represent the incorrect attachment. It is notable that tuples that represent correct attachments are highly represented by the incorrect tuples.

cand-p: depicts the probability of the candidate to be followed by the preposition.

$$cand\_p(v,p) = \frac{c(v,p)}{c(v) \cdot c(p)}, \; cand\_p(n,p) = \frac{c(n,p)}{c(n) \cdot c(p)}$$

cand-n2: the conditional probability of the candidate given n2 in the same context (a window of 5 words). The intuition to this feature can be explained using the pizza with anchovies examples. Anchovies are a popular pizza topping, and therefore are expected to frequently appear next to pizza. The relation between the pizza and the anchovies can be described without the preposition "with". For example – "I asked the vendor to put extra anchovies on my pizza".

$$cand\_n_2(v \mid n_2) = \frac{c(v,n_2)}{c(n_2)} \;\; , \; cand\_n_2(n \mid n_2) = \frac{c(n,n_2)}{c(n_2)}$$

Analysis of sample sentences from the BNC corpus

So far we have only used the unambiguous counts. The unambiguous counts cannot be used in a straightforward way, since each "hit" on a tuple supports the attachment to the candidate represented in the tuple which is not true for the unambiguous case. We, therefore, save additional information which includes the sentences that contain an ambiguous case induced by each tuple. Since, for some tuples, we have many examples, we randomly choose 100 sentences out of all the examples.

We extract the following features:

parser-vote-ratio: Each sentence is being parsed. We calculate the average decision of the Stanford Parser. Since we try to disambiguate cases where one of the attachments is necessarily wrong, regardless of the context (as opposed to "I saw and man with a telescope", where both attachments make sense), the incorrect combination of "candidate + p + n2" is not likely to appear as a valid attachment. There is additional discussion of the parser vote feature in Section 3.4.2

amb-count-ratio: is defined the same as the count-ratio, except that c is defined in a different way for tuples counts:

$$c\_amb(x) = (c(x) \cdot att\_rate(x) + \lambda \sum_{i \in e(x)} c(i) \cdot att\_rate(i)) \cdot ,$$

$att\_rate(x)$ = #attachments to the candidate represented in x/ #of sentences.

amb-relatedness: Another analysis on the ambiguous examples is to check the other attachment candidate in the ambiguous sentence. We follow the intuition that if for instance, a noun and a pp are found together several times, but the pp should be attached to the verb, then all of the verbs in this examples should be semantically related, and the same applies for a verb candidate and a pp. This feature measures the relatedness of the other candidates for each tuple, in respect to the noun candidate and to the verb candidate, and calculates the ratio between the two measures.

backoff-prob: We adapt Collins and Brook's backoff probability formula([4]) to use the counts of both unambiguous and ambiguous instanced or each tuple.

We define this to be the sum of the ambiguous and unambiguous instances tuple.

$$f(x) = c(x) + c\_amb(x)$$

The backoff algorithm to calculate $p(1 | v, n_1, p, n_2)$ (1 represents noun attachment) works as follows:

If $f(v,n_1,p,n_2)>0$:

$$p(1|v,n_1,p,n_2)=\frac{f(1,v,n_1,p,n_2)}{f(v,n_1,p,n_2)}$$

Else if $f(v,n_1,p)+f(v,p,n_2)+f(n_1,p,n_2)>0$:

$$p(1|v,n_1,p,n_2)=\frac{f(1,v,n_1,p)+f(1,v,p,n_2)+f(1,n_1,p,n_2)}{f(v,n_1,p)+f(v,p,n_2)+f(n_1,p,n_2)}$$

Else if $f(v,p)+f(n_1,p)+f(p,n_2)>0$:

$$p(1|v,n_1,p,n_2)=\frac{f(1,v,p)+f(1,n_1,p)+f(1,p,n_2)}{f(v,p)+f(n_1,p)+f(p,n_2)}$$

Else: if $f(p)>0$, $p(1|v,n_1,p,n_2)=\frac{f(1,p)}{f(p)}$, otherwise $p(1|v,n_1,p,n_2)=1$ (default)

We decide for the noun attachment if $p(1|v,n_1,p,n_2)>0.5$, otherwise it is a verb .

3.4.1.2 Realizations from FrameNet

Based on FrameNet realizations for each verb, we compiled a list of verbs + prepositions that appear together. This is not a complete list, and even if the preposition exists as a realization, it doesn't mean that it will always indicate verb attachment. For example: "I eat Hummus in a pita" as opposed to "I eat hummus in a restaurant" (first is noun attachment, second is verb attachment). FrameNet provided additional information for each realization. For the lexical entry "eat.v", we see that the proposition "in" refers to a place frame entity. In the previous example, this helps distinguish between "Hummus in a pita" and between "Hummus in a restaurant", since a restaurant is a place and therefore the frame fits. We still don't use any semantic frame information since attaching the frame to the sentence and determining each frame element is a difficult task that is still being studied, and can only add more noise to the system.

3.4.2 Quadruplets + Sentence Features

POS of the verb

As previously mentioned, we lemmatize all verbs as part of the preprocessing on the data. However, the POS of the verb may hold some valuable information to the disambiguation process. The motivation for this assumption can be found in the following example: he was brought to school by his dad. Usually, the preposition "by" will not accompany the verb "bring". However, the passive form of the verb indicated that a verb attachment is possible.

Stanford parser vote:

The Stanford parser, by default, returns a single parse tree with its score, but it can easily be modified to return the top K parse trees. Our assumption is that in a case of a real ambiguity, we will have several trees with noun attachment, and several others with verb attachment. We assume that in most cases, most of the parse tree will have the same attachment. The first tree is sometimes chosen because of other grammatical reasons, and it won't necessarily choose the correct PP attachment. When looking at the parse tree, we are only interested in the PP and its attachment. Trees with attachments that match neither n1 nor v are ignored.

The feature calculates the difference between the noun and verb attachment. We did not use the Stanford parser score in the process, since this score is calculated based on many other factors beside the PP attachments.

In addition to the majority feature, we also use the first parse tree (the actual output of the parser) as a feature, since in most cases we get a correct attachment.

3.4.3 Quadruplets and Context Features

Context Feature:

In some cases, only contextual knowledge can properly disambiguate a PP attachment. For the example "I saw a man with a telescope", we need to determine who has the telescope. Sometime this can be determined within the sentence ("I saw a man with my telescope" vs. "I saw a man with a telescope in his hand"). In other cases, we need to look for clues in the previous or the following sentences ("I saw a man with a telescope…. Later he packed his telescope and left the place").

Such ambiguity resolution can be very complicated. For the telescope example, we need to determine who has the telescope. Having a possessive pronoun attached to the word "telescope" simplifies the problem, but sometimes there are more complicated cases. The sentences "I saw a man with a telescope in his hand" is more complicated since the possessive pronoun is not attached to the telescope, and for "I saw a man with a

telescope. He was holding it with both hands" we already need anaphora resolution as well as the knowledge that “hold” means "clasps".

Clearly, for other prepositions, possession is irrelevant and we need to solve a different problem ("I've been looking for a trip for the entire month"), but even for "with" the relation is not always possession ("I ate pizza with friends").

We choose the following approach to handle the problem. For each attachment, we check for common appearances within k sentences from the analyzed sentence. To support noun attachment, we wish to find appearances of n1 + n2 in another sentence. To support verb attachment, we wish to find a common appearance of the subject of the verb + n2 (for "I ate pizza with friends" we check I + friends).

In the case that the subject or n1 are pronouns, we will expand the query with other forms. For instance, for I + friends we will also search for "my + friend" and also "me + friend".

3.5 Machine Learning

The RRR corpus as provided was already divided into training, development and test sets. For each sentence in each set, we generate a vector of features discussed in the Features Section 3. Next, we use Support Vector Machine (SVM) [26] to generate a predictive model that in turn will be used to classify the test sample and predict the correct classification - noun or verb. Figure 3.2 demonstrates this whole process.
As a machine learning tool, we use the Weka [13], an open source tool developed by the machine learning group from the University of Waikato. We used a polynomial kernel, and the parameters that are required for the training were determined using 2-fold cross validation.

4. Results

RRR dataset:

We have recreated Collins and Brook's back off model [4] on the RRR dataset with the accuracy of 84.3, which is slightly but not significantly lower than their reported accuracy (84.5%).

We then implemented our own version of the back off mode, using counts from BNC, as explained in Section 3.4.1.1, both with semantic expansions and without them.

We tried two approaches on this dataset: the first one is using SVM with a set of features, previously described, and the second one is MLE using the backoff-prob described in Section 3.4.1.1. The results are presented in Table 4.1

Description	accuracy
Most likely for each preposition	72.2
Maximum entropy, words & classes (Ratnaparkhi et al., 1994)	81.2
Maximum-Likelihood based (Collins and Brooks, 1995)	84.5
Nearest-neighbor (Zhao and Lin, 2004)	86.5
Maximum-Likelihood – counts on untagged corpus (*)	81.4
Maximum-Likelihood – counts with expansion on untagged corpus (*)	81.1
SVM using quadruplets features only (*)	80.3

Table 4.1: Accuracy of PP-attachment ambiguity resolution on the RRR dataset (our results in bold) are marked with *.

The accuracy of our methods is lower than the results of previous work on this task (Table 4.1). The back-off model proves to be the most accurate among the models we've tried, but the semantic expansions we added to the queries degraded the results, although not too drastically. Building our statistical model upon the counts on the untagged corpus is clearly exposed to some noise. Our definition of the un-ambiguous and ambiguous hits in the BNC are relatively tight and precise, but the drawback of our

method is the recall – the inability to identify correctly all such examples. Stanford vote that was used on the ambiguous examples is also clearly not accurate enough.

WSJ-sent dataset

For this data set, we also ran the Collins and Brook's back off model to use it as a baseline. In addition, we've conducted three sets of tests – quadruplets features only, quadruplets and the sentence, and quadruplets, sentence and context. The results are presented in Table 4.2.

ALGORITHM	ACCURACY
Collins and Brooks backoff	85.3
Maximum-Likelihood – counts on untagged corpus (*)	82.5
Maximum-Likelihood – counts with expansion on untagged corpus (*)	82.1
SVM – only quadruplets features (*)	79.3
SVM – quadruplets and sentence features (*)	83.3
SVM – quadruplets, sentence and context features (*)	82.5

Table 4.2: Accuracy of PP-attachment ambiguity resolution on the RRR dataset (our results in bold) are marked with *.

The best configuration was the one that didn't use the context features, and therefore indicates that the context, while valuable in some cases, negatively impacts the results. Stanford majority feature, on the other hand, proves to be significant when out of 6078 examples in the test set, Stanford parser attached correctly 4354 instances (71.63%), and the majority vote was correct for 4657 instances (76.62%). An example for a quadruplet shows that we got the majority votes correctly, but the first parse tree is incorrect is <bow,bidding,for,NAMEORGANIZATION> in the sentence "New England Electric System bowed out of the bidding for Public Service Co. of New Hampshire, …". The correct attachment appears in the second parse tree, but not the first one.

Errors analysis on best configuration of the SVM algorithm

- <be, executive, for, years> in the sentence: He had been a sales and marketing executive with Chrysler for 20 years

The correct attachment is an adverbial one, however our classifier assigns a noun attachment. Stanford parser attaches the PP correctly, while the majority votes for noun attachment (7 to 3). However, when we examine all parse trees, the first 3 trees (with the highest scores) attach the PP to the verb. The scores are similar, so in this case they wouldn't have helped.

The combination "be for year" (v+p+n2) appears 2429 times in the BNC corpus (without expansions), while "executive" for years appears only 3 times. The lemma "be" appears in the corpus 3918903 times, while executive appears 7825 times. Overall, the relative frequency of "be for year" is lower than "executive for year".

- <lift, ceiling, on, debt> in the sentence: The federal government suspended sales of U.S. savings bonds because Congress hasn't lifted the ceiling on government debt.

The correct attachment is to the noun. However, our classifier chose verb. Stanford parser attaches correctly, but the majority vote is slip (5 trees with each attachment type).

(import, ban, on, use) in the sentence: In July , the Environmental Protection Agency imposed a gradual ban on virtually all uses of asbestos.

The correct attachment is noun. This is an example of a sentence where the PP is in-correctly attached by the Stanford parser. When checking the majority vote, we get a tie – 5 trees attached to the noun and 5 trees attached to the verb. Our classifier attaches the PP correctly, since the combinations "ban use" and "ban on use" are more common than "impose on" and "impose on use".

- <succeed, NAMEPERSON ,on, board> in the sentence: They succeed Daniel M. Rexinger , retired Circuit City executive vice president , and Robert R. Glauber , U.S. Treasury undersecretary , on the 12-member board.

The correct attachment is to the verb. This is another example where our classifier succeeds while the Stanford parses fails to correctly attach the PP. All 10 parse trees that the Stanford Parser generate, attach the PP to the noun. However, the tuple (NAMEPERSON, on, board) is much more frequent than (succeed, on, board). Without semantic expansions, (NAMEPERSON, on, board) appears 65 times in the BNC corpus. Adding semantic expansions results in 179 hits. On the other hand, (succeed, on, board) appears only once, and twice with expansions. Added to the fact the board is more frequent in the corpus than succeed, the frequency feature strongly indicates that the correct attachment is to the noun.

- <reflecting, weakening, in, economy> in the sentence: Common wisdom suggests a single-digit rate of growth , reflecting a weakening in the economy and corporate profits .

The correct attachment in this example is to the verb. One of the sentences where the tuple “is the economy” was detected is “If we stick with 15 per cent rates until the Budget, the prospects for positive growth in the economy do not look too good” is attached to the noun (growth). The nouns growth and weakening are semantically related, and thus this this example supports noun attachment for the original tuple. Another supporting example can be found in the sentence “The cuts had been agreed without the collapse of the government or immediate catastrophe in the economy”. Also, “in the economy” is attached to “immediate catastrophe” which is semantically related to “weakening”.

An interesting BNC example sentence is “That would cause a sharp slowdown in the economy next year , but could also herald an improvement in the UK 's record trade deficit”. Stanford Parser attaches “in the economy” to the verb “cause” instead of slowdown. Had the PP been attached correctly, we would have gotten another support for the noun attachment, since slowdown and weakening are related. Nevertheless, when we compare the verb “use” to the verb “reflecting” from the original tuple, there is no connection between them. Therefore, even if in this example the attachment is to the verb, it does not support a verb attachment in the original tuple.

- <provides, service, in, states> in the sentence: Alltel , which provides local telephone service in 25 states , said it exercised its right of first refusal following an offer from an undisclosed third party to acquire the majority position in the franchise.

The correct attachment is to the verb, however, the classifier chooses to attach the PP to the noun. Applying the BNC example sentences feature, we came across the sentence “he ought to have had his own being a war baby and a cog in the welfare state” where “in the welfare state” was attached to the noun cog by the parser. The nouns “cog” and “service” are semantically related (through the sense “an act of help or assistance” for service and “a subordinate who performs an important but routine function” for cog), and therefore support noun attachment, while sentences with verb attachment contain verbs that are not semantically connected to “provide”.

5. Discussion

One of the main conclusions from this work, and from reviewing previous work is that a new benchmark is evidently needed. In several papers ([21], [6]), new datasets was created, mostly in order to align the quadruplets with their original sentences, but also to add more of the train/test examples and to correct the inaccuracies that were detected over the years in the RRR dataset (see Section 3.1). As previously mentioned, we also created a dataset of our own in order to overcome the flaws in the existing benchmark, but our obtained results cannot be compared to other reports. We assessed the quality of our data set using the Collins and Brooks back-off model, since it was done in [21] on their newly created data set. They report a bit higher accuracy, but this only emphasizes the necessity for the development and usage of a single new benchmark for this task.

One major drawback of our proposed method is the extensive data preprocessing. Ideally, we need to extract all possible quadruplets, triplets and couples from the BNC, count them, and parse at least some of the sentences they we extracted from. For the purpose of our work, we searched the BNC only for the tuples we have extracted from the data set, yet it was time and resources consuming. The integration of such an algorithm within a real time syntactic parser will require an extensive preprocessing stage, and an efficient solution to hold all the calculated statistics and to access them quickly.

We sense that the original sentence holds more information that can be valuable to the disambiguation process, besides the quadruplets representing the PP and its candidates, for instance, the adjectives or adverbs or v, n1, n2. An example is if we suppose that n2 is the noun "machine". Knowing that the machine is a coffee machine (and not a sewing machine) might be beneficial to the disambiguation process.

The expansion of our original quadruplets was done using WordNet and distributional probabilities methods. In recent years, word2vec[19] became one of the leading techniques to represent words as vectors in a semantic space. Therefore, it should be definitely considered as a technique to extract the related words for words in our quadruplets.

Another feature worth trying would be using FrameNet frames to evaluate possible adverbial attachments. For the combination v + p, if the attachment is indeed adverbial,

then n2 should be an element of the semantic frame for v + p. Picking the correct frame involves the disambiguation of the verb, as does the match of n2 to the frame elements. Nevertheless, it can enrich the semantic knowledge that we aim to collect about each candidate + PP.

Our experiments showed that the context feature had a low correlation to the correct attachment. The sparseness of the queries within the context can be blamed, naturally, but also the basic design of this feature (solely relying on counts, which creates a bias toward common verbs). We still believe that this should be further explored, despite the fact that this features forces the redesign of the classic syntactic parsing assignment of a single sentence to a sentence with its context. The semantic relations that are embedded in each preposition are perhaps too different to be captured within the context using the same features. It might require some tailoring for each preposition (for instance, to disambiguate "with" we might look for possession).

6. Bibliography

[1] Altmann, G. (1985) The resolution of local syntactic ambiguity by the human sentence processing mechanism. In proceedings of the second conference on European chapter of the Association for Computational Linguistics.

[2] Agirre E, Baldwin T, Martinez D. (2008) Improving Parsing and PP attachment Performance with Sense Information. In proceedings of ACL-08: HLT, pages 317–325.

[3] Atterer M, Schütze H. (2007) Prepositional Phrase Attachment without Oracles. In Computational Linguistics,33(4), paged 469–476.

[4] Baldwin T. , Kordoni V. , Villavicencio A. (2009). Prepositions in Applications: A Survey and Introduction to the Special Issue. In Computational Linguistics, pages 119-149

[5] Baker C. F. , Fillmore C. J., Lowe J. B (1998). The Berkeley FrameNet Project. In proceedings of the 17th international conference on Computational Linguistics, pages 86–90.

[6] Bharathi A., Rohini U., Vishnu P., Bendre S.M, Sangal R. A Hybrid Approach to Single and Multiple PP Attachment Using WordNet. Natural Language Processing – IJCNLP 2005, pages 211-222

[7] Brill E, Resnik. P. A. (1994) Rule-Based Approach to Prepositional Phrase attachment Disambiguation. In proceedings of the 15th conference on Computational Linguistics,pages 1198–1204.

[8] Budanitsky A, Hirst G (2006). Evaluating WordNet - based Measures of Lexical Semantic Relatedness . In Computational Linguistics Volume 32 Issue 1 Pages 13-47.

[9] Burnard, L. 2000. Reference Guide for the British National Corpus. Oxford University Computing Services.

[10] Charniak, Eugene, et al. BLLIP 1987-89 WSJ Corpus Release 1 LDC2000T43. DVD. Philadelphia: Linguistic Data Consortium, 2000

[11] Collins M , Brooks J. (1995) Prepositional Phrase Attachment through a Backed-Off Model. In third Workshop on Very Large Corpora.

[12] Finkel J.R., Grenager T., Manning C. (2005) Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370.

[13] Hall M., Frank E, Holmes G, Pfahringer B., Reutemann P, Witten I. H. (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[14] Hindle D, Rooth M. (1993) Structural Ambiguity and Lexical Relations. In Computational Linguistics, 19, pages 103–120.

[15] Kummerfeld J. K, Hall D, Curran J. R, Klein D. (2012) Parser Show-down at the Wall Street Corral: An Empirical Investigation of Error Types in Parser Output. In proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 1048-1059

[16] Klein D. , Manning C. D. (2003). Accurate unlexicalized parsing. In proceedings of the 41ST Annual Meeting of the ACL, pages 423–430

[17] Lesk M. (1986) Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In SIGDOC '86 Proceedings of the 5th annual international conference on Systems documentation, pages 24 - 26

[18] Marcus M.P, Santorini B, Marcinkiewicz M. A. Building A Large Annotated Corpus of English: The Penn Treebank. In Computational Linguistics 19, pages 313-330.

[19] Mikolov T., Sutskever I., Chen K., Corrado G., and Dean J. Distributed Representations of Words and Phrases and their Compositionality. In proceedings of NIPS, 2013

[20] Nakov P, Hearst M. (2005) Using the web as an implicit training set: application to structural ambiguity resolution. In proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing Pages 835-842.

[21] Oleanu M, Moldovan D (2005). PP-attachment disambiguation using large context. In proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 273–280.

[22] Panel p, Lin d. (2000) unsupervised approach to prepositional phrase attachment using contextually similar words. In proceeding ACL '00 Proceedings of the 38th Annual Meeting on Association for Computational Linguistics Pages 101-108.

[23] Ratnaparkhi A. (1998) Statistical Models for Unsupervised Prepositional Phrase Attachment. In proceedings of the 17th international conference on Computational linguistics(COLING '98), pages 1079 -1085.

[24] Ratnaparkhi A, Reynar R, Roukos S. A (1994) maximum entropy model for prepositional phrase attachment. In proceedings of the ARPA Human Language Technology Workshop, pages 250–255.

[25] Stetina J, Nagao M. (1997) Corpus based PP attachment ambiguity resolution with a semantic dictionary. In proceedings of the Fifth Workshop on Very Large Corpora, pages 66–80.

[26] Vapnik V , Cortes C (1995). Support Vector Networks. In Machine Learning, vol. 20, pages 273-297.

[27] Zhao S, Lin D. A nearest-neighbor method for resolving PP-attachment ambiguity. In proceedings of the First International Joint Conference on Natural Language Processing (IJCNLP-04), pages 545–554, 2004

אוניברסיטת תל-אביב TEL AVIV UNIVERSITY

הפקולטה למדעים מדוייקים על שם ריימונד ובברלי סאקלר
בית הספר למדעי המחשב על שם בלבטניק

הפגת עמימות בשיוך פסוקיות יחס באמצעות מידע סמנטי וקורפוס חיצוני גדול

חיבור זה הוגש כעבודת גמר לקראת התואר "מוסמך אוניברסיטה"
במדעי המחשב באוניברסיטת תל אביב על ידי
לנה דנקין

עבודה זו הוכנה בהדרכתו של
פרופ ' נחום דרשוביץ

סיוון , תשס"ה

תקציר

עמימות בשיוך פסוקיות יחס מתקבלת כאשר חוקי הדקדוק של השפה מאפשרים את קיומו של יותר ממועמד אחד לשיוך פסוקית יחס יחידה. למרות שלכל פסוקית יחס יכולים להיות מספר מועמדים לשיוך, רוב המחקר העוסק בבעיה זו התרכז במקרה יחיד שבו פסוקית היחס מופיעה מיד לאחר פסוקית שמנית, ואותה הפסוקית השמנית מופיעה אחרי פועל. מקרה זה מכסה את רוב המצבים שבהם המנתח התחבירי טועה בשיוך הפסוקית, וגם אנחנו עוסקים במקרה זה, אשר ניתן לראות בו גם בעיית סיווג בינארית, כלומר, הבחירה בין שיוך לפסוקית השמנית לבין הפועל.

בעבודה שלנו בדקנו שתי גישות עבור פתרון העמימות בשיוך פסוקיות היחס במשפטים בודדים. בגישה הראשונה, השתמשנו בלמידה מונחית (supervised learning) המבוססת על תכונות סמנטיות שיכולות לסייע בתהליך פתרון העמימות. גישה זו הצריכה עבודה עם מאגר טקסט גדול ומספר ארגונים (thesauri), ובנוסף שימוש בטכניקות מתקדמות של למידה חישובית. בגישה השניה, חיפשנו מידע נוסף בתוך מאגר המידע ממנו לקוחים המשפטים (משפטים שכנים ופסקאות) עבור כל משפט, בנסיון לחלק גם משם מידע אשר יכול לסייע בפתרון העמימות ומציאת השיוך הנכון.