Looking Beyond the Canonical Formulation and Evaluation Paradigm of Prepositional Phrase Attachment

Jonathan Schuman

A thesis
in
The Department
of
Computer Science

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science
Concordia University
Montreal, Quebec, Canada

December 2012

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Jonathan Schuman**
Entitled: **Looking Beyond the Canonical Formulation and Evaluation Paradigm of Prepositional Phrase Attachment**

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining commitee:

________________________________ Chair
Dr. Dhrubajyoti Goswami

________________________________ Examiner
Dr. Leila Kosseim

________________________________ Examiner
Dr. David Ford

________________________________ Supervisor
Dr. Sabine Bergler

Approved ________________________________
Chair of Department or Graduate Program Director

____________ 20 ______ ________________________
Dr. Robin Drew, Dean
Faculty of Engineering and Computer Science

Abstract

Looking Beyond the Canonical Formulation and
Evaluation Paradigm of Prepositional Phrase Attachment

Jonathan Schuman

Prepositional phrase attachment has long been considered one of the most difficult tasks in automated syntactic parsing of natural language text. In this thesis, we examine several aspects of what has become the dominant view of PP attachment in natural language processing with an eye toward extending this view to a more realistic account of the problem. In particular, we take issue with the manner in which most PP attachment work is evaluated, and the degree to which traditional assumptions and simplifications no longer allow for realistically meaningful assessments. We also argue for looking beyond the canonical subset of attachment problems, where almost all attention has been focused, toward a fuller view of the task, both in terms of the types of ambiguities addressed and the contextual information considered.

Acknowledgments

The completion of this thesis is due in large part to the enduring dedication and patience of my supervisor, Dr. Sabine Bergler, and her remarkable ability to work through, with, and around my often complete disregard for Gricean maxims. I am immensely grateful for her guidance and support in all aspects of my development in research.

I would like to thank the members of my defense committee, Drs. Leila Kosseim and David Ford for finding the time to read my thesis, and for their helpful comments and encouraging feedback.

The CLaC lab has provided a wonderfully quirky and supportive environment, and I thank everyone involved in making it so. In particular, I would like to thank Michelle Khalifé; the mini pep talks, coffee meetings, and gently nudging emails entitled "*Chou??*" may well have made the difference. I would also like to thank Julien Dubuc for lively discussion in all manner of geekery.

I extend my thanks to my sister, Tania Steinbach, for moral support and mediation efforts; to Celia and Angelo for convincing me long ago that it's okay to not want to just be an engineer; and to everyone who has put up with my (more-noticeable-than-usual) antisocial behavior over the past few years without writing me off completely.

Finally, I would like to express my gratitude for financial support from the Natural Sciences and Engineering Research Council of Canada, le Fonds de recherche du Québec—Nature et technologies, and the J.W. McConnell Family Foundation.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Prepositional phrase attachment is an important ambiguity resolution problem in syntactic parsing and semantic interpretation of natural language. It is a topic that has seen extensive coverage in the literature, but this coverage has generally been limited—particularly in the context of natural language processing—to a simplified view of the problem. This thesis looks at the simplifying assumptions conventionally applied in addressing and evaluating the problem of prepositional phrase attachment in natural language processing. We argue that attachment approaches must be formulated and evaluated more realistically if they are to offer any practical benefit in modern language processing tasks.

The words in this thesis will hopefully evoke somewhat unconventional thoughts in the reader about an already quite esoteric subject. That anyone, regardless of eloquence in writing or in speech, should hope to succeed at such a task—indeed, that we all do every day—is testament to the wonder of language. Our ability to understand each other hinges on a wide range of innate faculties and learned conventions operating at various levels of analysis. Attempts to understand these faculties, formalize these conventions, or re-implement this functionality in non-human machinery increasingly reveal just how remarkable a feat is language comprehension.

One characteristic of language that is seemingly incongruous with the general ease with which we understand each other is the pervasiveness of ambiguity at all levels of language analysis. We are keenly aware of many kinds of ambiguity in our daily use of language: the kinds that make comedy funny, imprecise writing difficult to follow, and doublespeaking politicians irritating. Our interest here, however, is the ambiguity that we usually resolve without even noticing. Whether we look at individual words, the structure of phrases and sentences, or how utterances relate to the world to which they refer, so much of what we say can be interpreted in so many different ways.

Consider the word *arms* in the following sentences:

(1.1) *a. The treaty brings us closer to a world free of nuclear arms.*
b. My gym membership brings me closer to a shirt filled with muscular arms.

The word itself is ambiguous, in that in either sentence *arms* can mean either weapons or human limbs, yet any literate human would immediately understand the implied sense in each case from the surrounding context.

Ambiguity in language can also occur at a structural level as it does in the following sentences:

(1.2) *a. John placed all the books on the top shelf.*

b. John read all the books on the top shelf.

The prepositional phrase *on the top shelf* in either sentence can relate to, or attach to, the noun *books* or the verbs *placed* or *read*, yet in each sentence most people would automatically select one interpretation without even noticing the alternative. In Example (1.2a), we interpret *on the top shelf* as an argument to *placed* because the act of placing requires that a location be specified. In Example (1.2b), we are likely to ignore the perfectly valid interpretation that the act of reading was carried out on the shelf because it is quite rare for people to read (or do anything else, really) on a shelf, but shelves are a common place to find books.

Even when all the words and the overall structure of a sentence are unambiguous, its interpretation may still be ambiguous as in the following:

(1.3) a. Maggie loves Greek food.
b. Maggie loves dog food.
c. Maggie loves junk food.

Here, the meaning of each word is clear and the structure of all three sentences is identical, yet their overall meanings are quite different. Our knowledge of the world makes it difficult for us to interpret *Greek food* as anything other than food made (at least traditionally) by Greek people, *dog food* as anything other than food made for dogs, or *junk food* as anything other than food made from nutritionally worthless ingredients. Nonetheless, each of these interpretations is linguistically available for each sentence—i.e. the food that Maggie loves could be made by, for, or out of Greeks, dogs, or junk.

These examples show ambiguities that humans can resolve so effortlessly that the casual reader would fail to recognize any ambiguity at all. Nonetheless, or perhaps because of this, such ambiguities are remarkable in the insights they provide into how language works and how we acquire and leverage our linguistic expertise and extensive world knowledge to understand each other.

In this thesis, we address prepositional phrase (PP) attachment ambiguity, the type of ambiguity seen in Example (1.2), within the practical context of natural language processing (NLP). Our goal is finding effective solutions to resolve attachment ambiguities arising in the course of automated parsing of natural language text, rather than insights into linguistic theory or human cognitive apparatus. As such, the conventions, assumptions, and paradigms that serve as our launching point come largely from early work in PP attachment by researchers working on the more general problem of automated parsing, and were arrived at in the context of the state of the art of parsing at that time. Automated parsing has improved dramatically since then, thanks in part to those early efforts in PP attachment, yet more recent work in PP attachment continues to confine itself to these early terms. The main contention of this thesis is that the evolution of automated parsing requires changing how we look at PP attachment. In particular, we posit that the traditional evaluation paradigm and problem formulation no longer provide any meaningful indication of the state of the art, and that future progress requires that they be properly assessed and revised.

1.1 Syntax and Parsing

A major reason for our ability to understand each other is that we all mostly adhere to the same structural conventions when constructing or interpreting sentences. We do not randomly jumble our words into sentences, but rather arrange them in particular ways to indicate how we intend them to relate to each other.

There are a great many words in the English language and new words are created continually. Yet all of them fall into a very small number of categories that define how they are used. These are variously called lexical categories, word classes, or parts of speech. The main parts of speech are nouns, verbs, adjectives, and adverbs. Roughly speaking, they can be defined as follows:

noun: refers to an entity, like a person (*Alice, girl*), place (*France*), or thing (*chair*), or to an abstract concept (*existentialism, happiness, bravery*)

verb: refers to an action (*run, think*)

adjective: qualifies or describes some property of a noun (*red* or *comfy* as in *red apple* or *comfy chair*)

adverb: qualifies a verb, adjective, or another adverb (*carefully* in *think carefully*; *very* in *very red apple* or *think very carefully*).

We depend on the arrangement and parts of speech of words to interpret sentences and determine who did what to whom. Consider these three sentences:

(1.4) *a. Alice ate mushrooms.*
b. Mushrooms ate Alice.
*c. *Alice mushrooms ate.*

The same three words appear in each sentence, yet because of their arrangements we interpret Example (1.4a) as a common dinner table scene or a plot element in a classic work of fiction, Example (1.4b) as a strange and horrific scene (perhaps from a sci-fi film about evil alien fungi invading Earth), and Example (1.4c) as meaningless, ungrammatical gibberish.

Of course, not all entities, actions, or qualities thereof can be expressed with a single word. Groups of words that function as a single unit in the structure of a sentence are called phrases. Phrases that function like nouns—referring to people, places, or things—are called noun phrases (NPs). Generally speaking, anywhere an individual noun can be used, a noun phrase can take its place without changing the overall structure of the sentence. For example the lone nouns *Alice* and *mushrooms* in Example (1.4), which are themselves NPs, can be replaced with more detailed NPs as follows:

(1.5) *a.* [$_{NP}$ *An abnormally tall Alice*] *ate* [$_{NP}$ *magic mushrooms*].
b. [$_{NP}$ *Carnivorous mushrooms from outer space*] *ate Alice.*

Similar relationships exist between verbs and verb phrases (VPs), adjectives and adjective phrases (ADJPs), and adverbs and adverb phrases (ADVPs). In this sense, the type of a phrase is determined by the lexical category of the words it behaves like or can replace. For the most part, a word belonging to this category is included within the phrase, providing the central meaning and/or behavior of the phrase as a whole. This word is called the head of the phrase. The heads in the following example phrases are underlined for illustration:

(1.6) *a.* [$_{NP}$ *an abnormally tall Alice*]
b. [$_{NP}$ *magic mushrooms*]
c. [$_{NP}$ *carnivorous mushrooms from outer space*]
d. [$_{VP}$ *think carefully*]

e. $[_{VP}\ \underline{\textit{eat}}\ \textit{mushrooms}]$
f. $[_{ADVP}\ \textit{very}\ \underline{\textit{carefully}}]$
g. $[_{ADJP}\ \textit{bright}\ \underline{\textit{red}}]$

A phrase can consist of a single head word, as do the NPs *Alice* and *mushrooms* in Example (1.4a-b); a grouping of words that function as a single unit, as do both NPs in Example (1.5a); or a grouping of phrases that function as a single unit, as does the highlighted NP in Example (1.5b), which is actually a grouping of the smaller NPs *carnivorous mushrooms* and *outer space*. In fact, each phrase within a sentence is both a functional grouping of smaller phrases or individual words, and a constituent of some higher-level phrase, which is why phrases are also often referred to as constituents. A sentence, then, is not merely a string of words or even phrases, but a hierarchical structure where individual words group into phrases which in turn group into larger and larger phrases, the largest of which is the sentence itself. It can be helpful to view this structure as a tree, as in Figure 1.1, which gives a syntactic analysis of Example (1.5b) in the form of a parse tree.

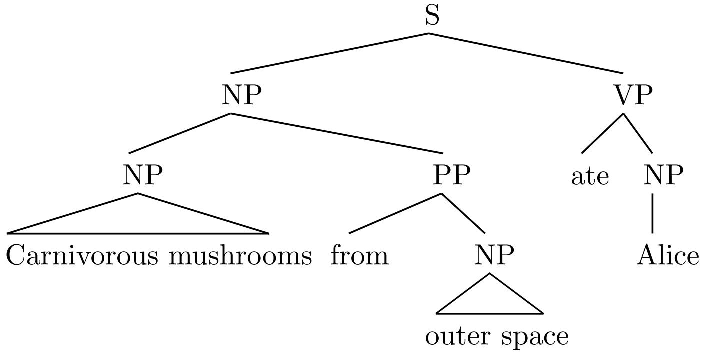


Figure 1.1: Parse tree depicting the structure of Example (1.5b)

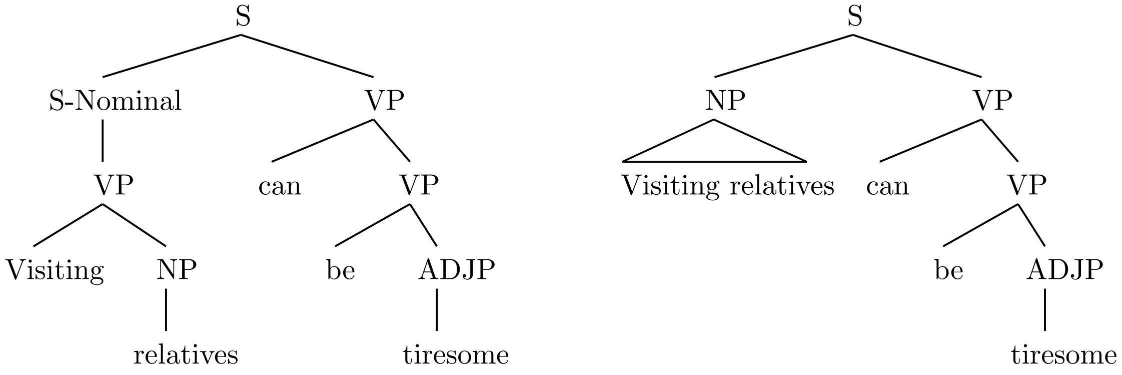
The structure of this hierarchy of phrases informs on the relations between words and how the sentence as a whole can be interpreted based on the meanings of its parts. In general, words or phrases within the same phrase are more closely related to each other than they are to other words or phrases. Moreover, specific structures indicate specific relations between constituents. At the coarsest level, the structure of a sentence indicates who did what to whom. Consider the parse tree in Figure 1.1. At the top level, the sentence takes the simplest form of English declarative sentences: an NP followed by a VP. Regardless of the internal structure of either of these phrases, in any sentence of this form, the NP specifies the subject and the VP specifies the event; that is, given any sentence of this form in active voice, in response to the question “Who did what to whom?” we can blindly take everything below the NP as the *who* and everything below the VP as the *what* (and possibly also the *whom*). In this particular sentence, the *who* is *carnivorous mushrooms from outer space* and the *what* is *ate Alice*. If we look at the internal structure of the VP, we can see that it takes the typical form of a simple VP: a verb followed by an NP. Again, regardless of the particular verb or internal structure of this NP, in any VP of this form the verb specifies the action (*what*) and the NP specifies the object of that action (*whom*).

While the possible internal structures of each type of phrase are well defined, determining the groupings of words and phrases for a given string of text is not always straightforward. A major challenge in parsing is dealing with syntactically ambiguous language—i.e. sentences that have more than one possible syntactically valid parse. Consider the following sentences, each of which has multiple syntactic interpretations, depicted in Figures 1.2–1.4:

(1.7) *a. Visiting relatives can be tiresome.*
b. Active dogs and cats need protein.
c. I saw the man with the telescope.

Depending on whether *visiting* is interpreted as the head of the gerundive verb phrase [$_{VP}$ *visiting* [$_{NP}$ *relatives*]] (as in Fig. 1.2a) or a modifier in the noun phrase [$_{NP}$ *visiting relatives*] (as in Fig. 1.2b), the meaning of Example (1.7a) can either be that I find it tiresome to visit relatives or that I find them tiresome when they visit me, respectively. In Example (1.7b), the conjunction can have wide or narrow scope—i.e. the statement can apply to active dogs and active cats (excluding lazy cats, as in Fig. 1.3a), or to active dogs and all cats (active, lazy, or otherwise, as in Fig. 1.3b). The telescope in Example (1.7c) can be an instrument I use to see the man in question, or something he possesses—perhaps distinguishing him from other men I may have seen—depending on whether the prepositional phrase attaches to the verb *saw* (as in Fig. 1.4a) or the noun *man* (as in Fig. 1.4b).

A cursory glance at Figures 1.2–1.4 may suggest that these three examples are similar variations on the general problem of syntactic ambiguity. In each case, a choice must be made from among multiple possible parses, each leading to rather different interpretations of the sentence. But these are not as similar as they seem. Figure 1.4 provides a classic example of prepositional phrase attachment, our topic of inquiry. Prepositional phrases are not quite like other phrases, and disambiguating their attachment is one of the most prominent challenges to accurate syntactic analysis of natural language.



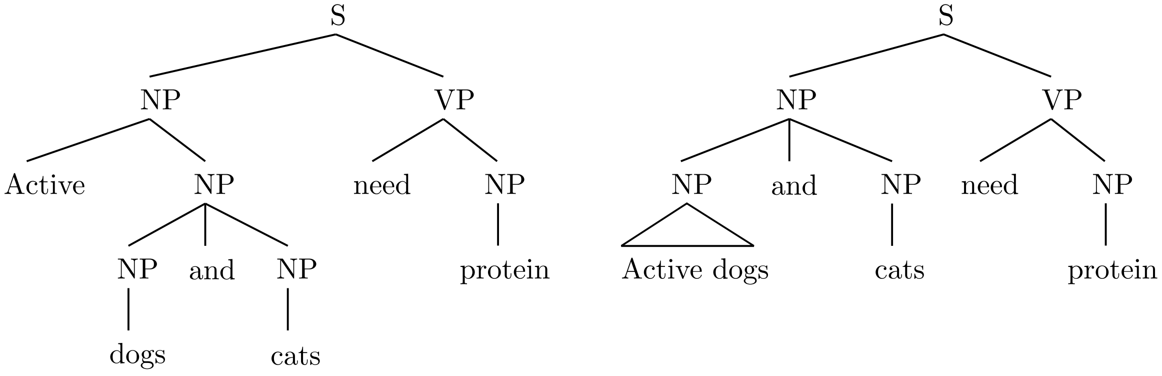
a. gerund construction

b. participle construction

Figure 1.2: Two possible syntactic analyses for an ambiguous subject

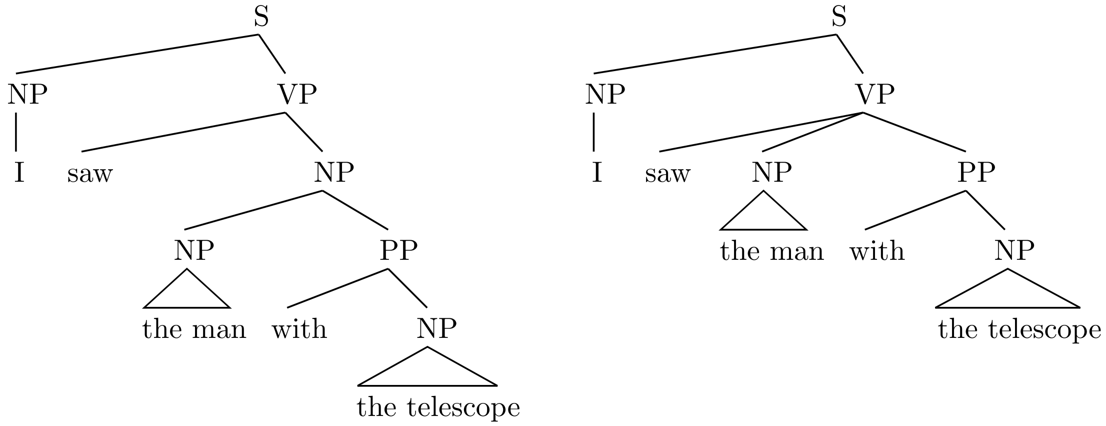
1.2 Prepositional Phrase Attachment

Why then is prepositional phrase attachment so difficult? In short, prepositions and their phrases are terribly versatile things; they can be used in many ways, often introducing different forms of ambiguity. We loosely described a phrase, in the previous section, as a group of words that functions as one unit, taking the behavior of its head. Prepositional phrases are the exception to this rule of thumb. Each PP is indeed a group of words that functions as a unit, but its behavior does not mimic that of a lone preposition. In fact a



a. wide scope b. narrow scope

Figure 1.3: Two possible syntactic analyses for an ambiguous conjunction



a. noun attachment b. verb attachment

Figure 1.4: Two possible syntactic analyses for an ambiguous PP attachment

PP can take on many quite different behaviors. They can function as arguments to verbs and nouns, as in Example (1.8a) and (1.8b), respectively.

(1.8) a. *John placed the books on the top shelf.*
b. *John is a student of linguistics.*

They can also take on the role of other types of constituents. For example, PPs can be functionally equivalent to adverb phrases, as in Example (1.9a), postnominal adjective phrases,[1] as in Example (1.9b), and predicative adjective phrases, as in Example (1.9c).

(1.9) a. *I saw John* $\left\{\begin{array}{l}[_{PP}\ \textit{on Tuesday}]\\ [_{ADVP}\ \textit{yesterday}]\end{array}\right\}$.
b. *I bet on the team* $\left\{\begin{array}{l}[_{PP}\ \textit{with the best record}]\\ [_{ADJP}\ \textit{most likely to win}]\end{array}\right\}$.

[1] While adjective phrases are not often used postnominally in English, many prenominal adjectives can be expressed equivalently with a PP behaving as a postnominal ADJP (e.g. *scotch whisky/whisky from Scotland*, *the bearded man/the man with a beard.* Thus PPs behaving as postnominal ADJPs are much more common than actual postnominal ADJPs.

c. *The trees are* $\left\{\begin{array}{l}[_{PP}\ \textit{on fire}] \\ [_{ADJP}\ \textit{ablaze}]\end{array}\right\}$.

Determining the functional behavior of a PP is difficult without knowing its attachment, and vice versa.

Prepositions also differ from the four main categories of words in that they play a functional rather than lexical role; prepositions do not refer to objects or actions as do nouns and verbs, but to relations between them. The main topic of the noun phrase *an abnormally tall Alice* is specified by its head, *Alice*, and we can similarly say that the verb phrase *eat mushrooms* is about eating. By this reasoning, we can say that the prepositional phrase *from outer space* should be about *from*-ness, which, while true, does not describe the whole picture. A prepositional phrase describes part of a relation between two words or phrases. Looking at a PP alone, we may be able to determine the type of relation, or semantic role, involved, in this case an `ORIGIN` relation, and one of the participants, in this case *outer space*. In order to see the full relation and determine what the PP is really about, we must determine where it attaches—i.e. what word or phrase is meant to replace x in the relation $\texttt{ORIGIN}(x, \text{outer space})$. In the sentential context of Example (1.5b), x is clearly the carnivorous mushrooms, but prepositional phrase attachments are not always so obvious.

Determining the semantic role of a PP can also be less than obvious. Prepositional phrases are capable of expressing a wide spectrum of different types of relations, from rather concrete relations of location and time to more abstract relations of manner (see Figure 1.5 for some examples), and there is no one-to-one correspondence between prepositions and the relations they can express; each preposition can express many different relations, and most relations can be expressed with any number of different prepositions.

Both determining semantic roles and finding attached arguments are integral parts of extracting relations like `ORIGIN`(mushrooms, outer space) from natural language text and understanding the text in general. The two tasks are also complementary: knowing the semantic role in question helps in finding attachments, and vice versa. However, both tasks constitute significant NLP challenges in their own right, and a full, integrated treatment of both is beyond the scope of our current discussion, if not the current state of the art. Our concern here is mainly with the attachment task in isolation, though we will consider semantic role information in as much as it proves helpful in deciding attachments.

The diversity of prepositional phrases, both in terms of the relations they can express and the functional roles they can fill, represents a significant challenge to their attachment. A great deal of context can be required to accurately attach PPs, and different types of contextual cues may be essential, irrelevant, or even misleading depending on the type of attachment ambiguity at hand. Consider again the classic example of PP attachment

location:	*in the car, on the table, at home, by the river*
time:	*on Tuesday, in one hour, before dinner, by tomorrow*
instrument:	*cut* [$_{PP}$ *with a knife*], *paint* [$_{PP}$ *with a brush*]
accompaniment:	*hiking* [$_{PP}$ *with dogs*]
manner:	*in a hurry, with great urgency*
purpose:	*call* [$_{PP}$ *for help*]
agent:	*elected* [$_{PP}$ *by the people*]

Figure 1.5: A sampling of the variety of relations expressible with prepositional phrases

ambiguity:

(1.10) I saw the man with the telescope.

Our task is to determine if *with the telescope* specifies an instrument used to see the man, or if it qualifies *the man* to distinguish him from other people I may have seen. The two alternatives involve either the `INSTRUMENT` or `POSSESSION` relations. As such, resolving this attachment is roughly equivalent to correctly determining the semantic role, a difficult task as we have stated because of the lack of one-to-one correspondence between prepositions and semantic roles and the need for context to make a correct decision. But even if perfect (or at least adequate) semantic role labeling were available, many attachment ambiguities would still be quite difficult. Consider the following:

(1.11) I saw the man in the park.

Here, the PP *in the park* unequivocally expresses a `LOCATION` relation, yet this information does not facilitate attachment in the slightest. Without any additional context, it is just as reasonable to assume the park to be the location of the man or the act of seeing him.

It is worth noting that Example (1.10) and Example (1.11) are both syntactically and semantically ambiguous. That is, not only are there two valid structural analyses for each [those in Figure 1.4 for Example (1.10)], but the two meanings entailed by either structure both make sense. Indeed, the reason they are classical examples of the PP attachment problem is precisely because a reader can immediately see the semantic ambiguity, and thus the syntactic ambiguity is easily brought to light. Neither of the analyses in Figure 1.4 can be judged as correct or incorrect without further discourse context. Contrast this with the following sentences:

(1.12) a. I saw the man with a beard.
b. I saw the man with my own eyes.

The attachments here are semantically unambiguous. We know that beards cannot be used as instruments in seeing, therefore verb attachment is not semantically available for Example (1.12a). Similarly, we know that my eyes are generally not in the possession of anyone but myself and that if perchance they came to be, I certainly would lose the ability to visually observe their new possessor. As such, noun attachment is not semantically available for Example (1.12b). Still, in both these examples, the alternative interpretations are syntactically valid. That a reader may need prompting to see the ambiguity is because he or she, having knowledge of the meaning of words and the world to which they refer, is automatically calling on his or her semantic knowledge to resolve the syntactic ambiguity. The attachment approaches discussed in this thesis strive to realize precisely this kind of ambiguity resolution ability in machinery having little to no knowledge of the meaning of words or the world to which they refer.

The distinction between semantic and syntactic ambiguity is important in understanding what makes PP attachment such an important and challenging part of any automated syntactic analysis. A conscientious writer may try to avoid constructions that are obviously ambiguous (at the semantic level). As such, phrases like *visiting relatives* or even *saw the man with a telescope* may be avoided in favor of more precise formulations. But even the most meticulous writing would likely include syntactic ambiguities like those in Example (1.12). At the level of syntax, most uses of prepositional phrases are inherently and unavoidably ambiguous.

1.3 Canonical Form

The linguistics literature, and the natural language processing literature to an even larger degree, generally confines itself to a restricted subset of the PP attachment issue. Ambiguity resolution techniques also consider only a restricted subset of contextual features as relevant to disambiguation. In this section, we outline the simplifications and omissions that make up the conventional view of PP attachment.

1.3.1 Binary V/N Ambiguity

The most definitive aspect of the canonical PP attachment problem is that the ambiguity is between exactly two options: verb attachment or noun attachment. That is, the only PPs of interest are those with the type of ambiguity exemplified in Examples (1.10–1.12). The canonical form omits from consideration any PPs with multiple noun attachment candidates, as in Example (1.13a) below, multiple verb attachment candidates, as in Example (1.13b), or attachment candidates that are neither nouns nor verbs, as in Example (1.13c), where the PP attaches to the adjective *similar*.

(1.13) *a. I saw the* [$_{noun}$ *man*] *in the* [$_{noun}$ *park*] [$_{PP}$ *with the telescope*].
b. I [$_{verb}$ *saw*] *the man* [$_{verb}$ *feeding*] *birds in the park* [$_{PP}$ *with the telescope*].
c. Binoculars are [$_{adj}$ *similar*] *in function* [$_{PP}$ *to telescopes*].

The all but exclusive focus on this binary form of attachment ambiguity can be explained in part by the fact that its most common pattern of occurrence—where a verb, its object, and an ambiguous PP occur in direct succession—is the most common form of PP ambiguity overall.[2] Still, the vast majority of ambiguous PPs are not verb/noun (V/N) ambiguous PPs.

Ultimately, limiting the scope of inquiry to V/N ambiguities is a simplification. It has allowed us to make progress on the easiest part of a difficult problem, but it is unclear whether the lessons learned scale up to cases with greater ambiguity, as we discuss in Chapter 4.

1.3.2 Head-based Relationship

Another aspect of the canonical form concerns itself not with which ambiguities to consider, but rather what information to use in resolving the ambiguity. Many, though not all, approaches to PP attachment approximate attachment as a head-to-head relationship. Accordingly, only the head words of the relevant phrases are considered. These techniques ignore all other context, such as prenominal modifiers, and as such have no basis on which to distinguish between attachments like those in the following two examples:

(1.14) *a. I saw the man with my own eyes.*
b. I saw the man with blue eyes.

Again, the possessive prenominal modifiers in Example (1.14a) provide a strong bias toward the verb-attachment interpretation—a bias that is not present in Example (1.14b). Yet, in a canonical, head-only model of PP attachment, both attachment instances are identical; the context of each being the heads *saw*, *man*, *with*, and *eyes*.

[2] This is based on Mitchell's analysis of the 20 most frequent patterns of PP ambiguity in the Wall Street Journal corpus (2004, p. 152). We interpret any pattern where a verb, noun, and PP occur in direct succession to represent binary V/N ambiguity. In his notation, these patterns are: `[VG SNP PP]`, `[VG PP PP]`, and `[VG QP PP]`, all together accounting for 36.73% of ambiguities in his analysis.

1.3.3 Independence

The final constraint we will consider as part of the canonical form is the assumption that each PP attachment is independent of any other PP attachment. At first glance, this may seem an irrelevant issue when considering only binary ambiguity decisions. So far, our examples of binary attachment ambiguity have involved simple sentences with only one PP, but binary ambiguities can and do appear in more complex sentences. Consider the following examples:

(1.15) *a. Sam loaded the boxes* [$_{PP}$ *on the cart*] *into the van.*
b. Sam loaded the boxes [$_{PP}$ *on the cart*] *before his coffee break.*

In both sentences, attachment of the PP *on the cart* to either the verb *loaded* or the noun *boxes* is syntactically possible, yet the actual attachment in each sentence differs based on the context provided by the final PP.

Of course, PPs with a greater degree of ambiguity can be similarly affected by the PPs preceding or following it. They are also influenced by preceding PPs in that the latter generally introduce additional attachment site candidates. As such, assumptions of independence may be even more of a concern when dealing with more ambiguous attachments.

1.3.4 Notation

Given these simplifying restrictions, we represent the canonical task of prepositional phrase attachment as a binary classification decision between verb and noun attachment with a quadruple specifying the relevant lexical heads:

$$(v, n, p, n_c),$$

where v is the potential verb attachment site, n is the potential noun attachment site, p is the preposition, and n_c is the head noun of the prepositional complement. The ambiguity in Example (1.10) would thus be represented as $(saw, man, with, telescope)$. Where the correct attachment, A, is known—i.e. in training instances—it is prefixed to the tuple:

$$(A, v, n, p, n_c),$$

where A can be V for verb attachment or N for noun attachment.

Chapter 2

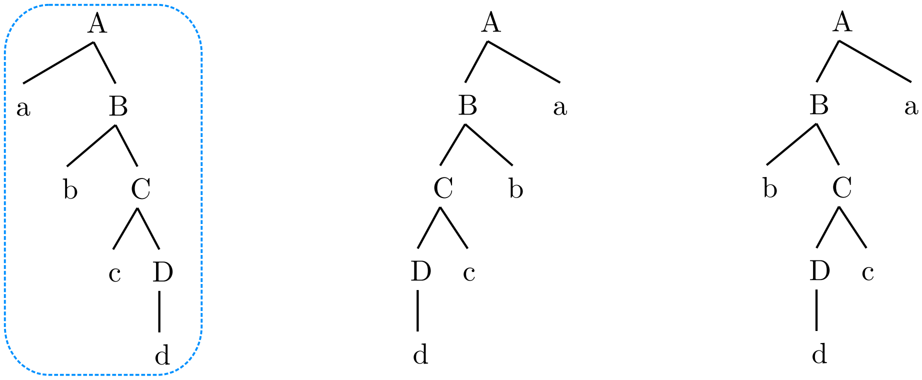
Attachment Techniques and Concepts

Accounts of human sentence processing can be categorized according to the degree of autonomy or interaction ascribed to syntactic processing with respect to higher-level processing. At the autonomous end of the spectrum [e.g. (Rayner, Carlson, and Frazier, 1983; Ferreira and Clifton, 1986)], modules are postulated to work largely in isolation, with information flowing sequentially and in mainly one direction. Here, the sentence processing mechanism would resolve structural ambiguities, including PP attachments, and arrive at a single syntactic analysis using only structural information. Semantic information would inform on syntactic analysis only when higher-level modules determine an initial analysis to be inconsistent with the current context, requiring an alternative parse.

The linguistics and psycholinguistics literature contains several proposals for structural principles employed by the human parsing mechanism to cope with syntactic ambiguity in the face of processing and memory limitations. One such strategy is the principle of right association (Kimball, 1973), or the similar principle of late closure (Frazier, 1979), which describes a preference to attach new constituents within the lowest open constituent, rather than a phrase higher up in the tree. It is meant to explain the frequency of right-branching parses in natural language sentences, as in Figure 2.1a, as well as the greater complexity perceived by human test subjects asked to process alternative structures, as in Figure 2.1b-c. Minimal attachment (Frazier, 1979) is another such strategy, describing a preference to attach new constituents using as few nodes and branches as possible.

At the other end of the autonomy/interaction spectrum, it is posited that humans process sentences incrementally and that a much more dynamic interaction between modules occurs. Here, the sentence processing mechanism would entertain multiple possible analyses of a sentence simultaneously, re-assessing the plausibility of each as more words or phrases from the sentence are understood, and as higher-level modules refine their contextual interpretations. Altmann and Steedman (1988) showcase a rich assortment of contextual and referential cues that inform on structural ambiguity decisions, suggesting that higher-level semantic and discourse knowledge is necessary to resolve these ambiguities. They give evidence for human parsing machinery that operates interactively with higher-level processing in an incremental fashion, rather than in isolation.

Narrowing in on the task of disambiguating PP attachments, Whittemore, Ferrara, and Brunner (1990) show neither right association nor minimal attachment account for more than 55% of attachments in a corpus of naturally occurring text. Instead, their study



a. right branching b. left branching c. center embeded

Figure 2.1: Branching structures

shows lexical preference—the tendency of certain verbs or nouns to prefer certain PPs, and vice versa—to be a much better predictor of attachment behavior than purely structural principles. Essentially all PP attachment work in natural language processing has focused on acquiring and applying such lexical information effectively.

This chapter outlines the major approaches along these lines. The primary objective here is to catalog useful concepts and ways of thinking about the problem of PP attachment in the context of NLP, rather than a critical assessment of any particular approach or system. As such, assessment is largely deferred to Chapter 3, where we examine evaluation concerns in depth. For our immediate purposes, it suffices to say that all of the approaches here outlined provide significant contributions to our understanding of the problem.

2.1 Lexical Association

Hindle and Rooth (1993) present the first treatment of automated PP attachment based on lexical statistics compiled from a large corpus of text. The idea behind their approach is that lexical preferences, already shown to be quite useful in predicting attachment, can be estimated by counting lexical co-occurrences. Here co-occurrence refers not necessarily to direct adjacency or proximity of two words, but to their relationship through prepositional attachment. That is, each instance of a specific preposition attaching to a specific verb or noun counts as a co-occurrence of the two words.

With no large collection of annotated data available, they obtained their co-occurrences counts from automatically generated partial parses of an unlabeled, 13 million-word sample of Associated Press news stories from 1989, using an iterative unsupervised approach. (We will discuss unsupervised attachment methods and unambiguous attachments more closely in Section 2.3.) The partial parses were generated using the Fidditch parser (Hindle, 1983), a deterministic parser that refrains from making attachment decisions (involving PPs or other constituents) where there is uncertainty. From these parses, they extract the heads of all noun phrases, together with the following preposition, and the preceding verb, if the NP is the object of that verb. Each such head-word triple is counted as an attachment site/preposition co-occurrence. In the case of ambiguous attachment, co-occurrence counts are determined using the following procedure:

1. Using the co-occurrences computed so far, if the lexical association score for the ambiguity (a score that compares the probability of noun versus verb attachment, as described below) is greater than 2.0, assign the preposition to verb attachment. If it is less than -2.0, assign the preposition to noun attachment. Iterate until this step produces no new attachments.

2. For the remaining ambiguous triples, split the co-occurrence count between the noun and the verb, assigning a count of .5 to the noun/preposition pair and .5 to the verb/preposition pair.

3. Assign remaining pairs to the noun.

The lexical association score is defined as the log likelihood ratio,

$$\mathrm{LA}(v, n, p) = \log_2 \frac{P(p|v)}{P(p|n)},$$

where the conditional probability of the preposition p attaching to the given verb v is computed from the co-occurrence frequencies determined in the procedure given above as follows:

$$P(p|v) = \frac{f(v,p) + \frac{f(V,p)}{f(V)}}{f(v) + 1}.$$

The conditional probability of the preposition p attaching to the given noun, n is similarly computed:

$$P(p|n) = \frac{f(n,p) + \frac{f(N,p)}{f(N)}}{f(n) + 1}.$$

The conditional probability formulae take into account attachment site/preposition co-occurrences, $f(v,p)$ or $f(n,p)$, normalized over the total frequency of the respective verb or noun, $f(v)$ or $f(n)$, regardless of attachments. Also factored into the computation are the total verb attachment frequency for a given preposition, $f(V,p)/f(V)$, and total noun attachment frequency, $f(N,p)/f(N)$. Here, $f(V,p) = \sum_v f(v,p)$, $f(V) = \sum_v f(v)$, $f(N,p) = \sum_n f(n,p)$, and $f(N) = \sum_n f(n)$. The total noun and verb attachment frequency terms are included to smooth out sparsity in the data. Where no co-occurrences have been observed for a particular verb/preposition pair or noun/preposition pair, the preposition's general tendency toward verb attachment or noun attachment, irrespective of particular lexemes, can still be used to inform on the decision.

Once calculated, the lexical association score indicates whether verb or noun attachment is more likely for a given triple (v, n, p). Positive scores indicate that verb attachment is more likely, while negative scores indicate noun attachment is more likely. A score of zero occurs when no evidence has been observed for either case—i.e. the preposition was not seen during training. Further, the magnitude of the score gives an indication of the certainty of the decision. A score of 1.0 indicates that verb attachment is somewhat more likely than noun attachment, whereas a score of 10.0 indicates a much stronger verb attachment likelihood.

Further work on lexical-statistics-based attachment was heavily influenced by the release of a large PP attachment corpus (Ratnaparkhi, Reynar, and Roukos, 1994), henceforth the RRR corpus. Extracted from a preliminary version of the Penn Treebank (PTB) Wall Street Journal (WSJ) corpus (Marcus, Marcinkiewicz, and Santorini, 1993), the RRR corpus consists of roughly thirty thousand head-word quadruples, specifying the potential verb attachment site, potential noun attachment site, the preposition, and its complement, along with the correct attachment for each, as described in Section 1.3.4. Its release opened the field to anyone capable of running statistical learning machinery, without the need to get bogged down in the details of extracting information from or manipulating actual natural language texts or parse trees. In fact, PP attachment became a convenient test problem for comparing machine learning techniques, such as maximum likelihood estimation (Collins and Brooks, 1995), maximum entropy (Ratnaparkhi, Reynar, and Roukos, 1994) and log-linear (Franz, 1996) modeling, decision tree induction (Stetina and Nagao, 1997), boosting (Abney, Schapire, and Singer, 1999), Markov chains (Toutanova, Manning, and Ng, 2004), and support vector machines (Olteanu and Moldovan, 2005).

It would be infeasible, and of little benefit, to detail all attachment efforts based on the RRR corpus. However, one particular approach stands out for its intuitive simplicity. Collins and Brooks (1995) apply a maximum likelihood estimation (MLE) approach, in similar fashion to Hindle and Rooth, though the availability of annotated data affords them a more precise model. They also present a more fine-grained approach to handling sparsity.

Consider the task of deciding an ambiguous attachment as equivalent to determining the conditional probability that the noun is the correct attachment site, given a quadruple as described above, $P(N|v, n, p, n_c)$.[1] This conditional probability can be estimated from the training data as,

$$P(N|v, n, p, n_c) = \frac{f(N, v, n, p, n_c)}{f(v, n, p, n_c)}$$

So, for example, to disambiguate $(saw, man, with, telescope)$ extracted from Example (1.10), we would divide the number of occurrences of $(saw, man, with, telescope)$ in our training data representing noun attachment by the total number instances, regardless of attachment.

But what if there are no such instances in the training data? A model is not particularly useful if it can only make decisions it has previously seen, without any ability to generalize. Collins and Brooks' solution is to use partial information when necessary. If the likelihood of an attachment cannot be determined for a quadruple due to sparsity of the training data, a backed-off model is applied using the attachment likelihoods of the three subset triples (v, n, p), (v, p, n_c), and (n, p, n_c)—the preposition is never omitted as it was determined to contribute the most to attachment decisions. This smoothing technique is applied repeatedly, trying 4-, 3-, 2-, and 1-tuples until an attachment likelihood can be determined. The full procedure is given formally in Algorithm 2.1.

Basic morphological preprocessing can be applied to quadruples to further reduce sparsity. Collins and Brooks achieved their best results using tuples where the verb was replaced by its lemma (a canonical form of the verb representing all of its morphological variants), the preposition and verb were transformed to lower case, and basic patterns in both nouns (n and n_c) were detected (e.g. all numbers were replaced by `NUM` and all nouns beginning with an upper case letter followed by at least one lower case letter were replaced with `NAME`). Such preprocessing affords the model a basic level of generalization, allowing for example the following otherwise disparate training instances:

(2.1) a. *Give your money to Alice.* ⇒ $(V, Give, money, to, Alice)$
b. *Giving my money to Bob wasn't easy.* ⇒ $(V, Giving, money, to, Bob)$

[1] Note that because we are concerned here with a binary decision, it makes no difference whether we frame the decision in terms of the conditional probability of noun attachment or verb attachment, since $P(V|v, n, p, n_c) = 1 - P(N|v, n, p, n_c)$.

c. She gave the money to Charles. ⇒ $(V, gave, money, to, Charles)$

to be generalized as $(V, give, money, to, \texttt{NAME})$. These training instances can then be applied as evidence for disambiguating an unseen quadruple like $(gave, money, to, Denise)$, which might otherwise require backing off to lower-order tuples.

Algorithm 2.1 Collins & Brooks' backed-off estimation procedure

```
procedure ESTIMATE-NOUN-ATTACHMENT-PROBABILITY($v, n, p, n_c$)
    if $f(v, n, p, n_c) > 0$ then
        $P(N|v, n, p, n_c) \leftarrow \frac{f(N, v, n, p, n_c)}{f(v, n, p, n_c)}$
    else if $f(v, n, p) + f(v, p, n_c) + f(n, p, n_c) > 0$ then
        $P(N|v, n, p, n_c) \leftarrow \frac{f(N, v, p, n_c) + f(N, n, p, n_c) + f(N, v, n, p)}{f(v, p, n_c) + f(n, p, n_c) + f(v, n, p)}$
    else if $f(p, n_c) + f(v, p) + f(n, p) > 0$ then
        $P(N|v, n, p, n_c) \leftarrow \frac{f(N, p, n_c) + f(N, v, p) + f(N, n, p)}{f(p, n_c) + f(v, p) + f(n, p)}$
    else if $f(p) > 0$ then
        $P(N|v, n, p, n_c) \leftarrow \frac{f(N, p)}{f(p)}$
    else
        $P(N|v, n, p, n_c) \leftarrow 1$
    end if
    return $P(N|v, n, p, n_c)$
end procedure
```

The backed-off model provides an intuitive framing of the attachment problem, but the contribution of this work does not end there. Another significant contribution comes from the experiments Collins and Brooks carried out to find the optimal thresholds for backing off—the minimum number of relevant 4-, 3-, 2-, or 1-tuples that the model requires to make an attachment decision without backing off to lower level. In many language modeling tasks low-count events are often smoothed over. The underlying conventional wisdom is that both events that occur very infrequently in training data and events that are entirely unseen in training are likely to be rare in reality. For a given rare phenomenon, the difference between occurring once or twice in a training set or not at all can be just as easily attributed to chance in sampling rather than any qualitative difference. Thus, to avoid disproportionate bias, many smoothing techniques ignore frequency counts below a given threshold, redistributing the probability mass to unseen events. For example, Google's corpus of n-grams (Brants and Franz, 2006), used for many language modeling tasks from statistical machine translation to speech recognition, includes only occurrences observed at least 40 times in the source text. In Collins and Brooks' backed-off model, smoothing over low-count events would be achieved by having a non-zero frequency threshold for back-off levels. Say for a given quadruple whose attachment we wish to decide, only a few equivalent quadruples occur in the training data. Conventional wisdom suggests that this may be too few data to make an accurate decision, and that we may be better off ignoring these quadruples and backing-off to triples instead. However, Collins and Brooks determined the optimal back-off threshold to be zero for all levels of backing off—i.e. that it is always better to use even a single available instance of a higher-order tuple rather than backing off to lower-order tuples.

2.2 Similarity-based Attachment

Smoothing techniques, like backing off to less specific models, and basic morphological processing (e.g. stemming) can alleviate sparsity in what would otherwise be exact string matching of quadruples. However, we should be able to exploit previously learned lessons in a much wider range of similar contexts. Take the following sentences, for instance:

(2.2) *a. John gave Mary a book about syntax.*
b. John gave Mary an article about syntax.
c. John gave Mary a poem about syntax.

Once we have determined that the PP *about syntax* in Example (2.2a) attaches to *book*, say through application of Collins & Brooks' method, it seems intuitive that the same frequency information and thus attachment decision should be applicable to Example (2.2b), given that books and articles are very similar things: written documents, which tend to have topics of focus, like syntax among others. The tremendous power of this way of reasoning is best highlighted in Example (2.2c). It is highly unlikely that any treebanked data anywhere could provide any instances of (*poem*, *about*, *syntax*). Yet, any human reader having considered the attachments in the first two sentences of Example (2.2) would immediately apply the same interpretation to the third, whether or not the reader has ever heard of poetry about such technical concerns, or can even imagine such an oddity.

But how can we possibly hope to assess and exploit similarity among words using machinery that has no notion of the real-world concepts and things to which these words refer? Here we outline approaches that attempt just that, using manually compiled resources or corpus statistics for semantic knowledge.

2.2.1 Lexicon-based Similarity

An obvious resource for semantic knowledge of words is the one most humans turn to when faced with unfamiliar terminology: the dictionary. Dictionaries allow us to understand unknown words by relating them to words and contexts we do know. This is a valuable asset when trying to compare and contrast the PP-modified nouns in Example (2.2), all of which are defined as forms of writing and thus similarly likely to be about syntax, or about any other field of inquiry.

In a pilot study, Jensen and Binot (1987) attempt to apply just such reasoning to the task of attaching a small subset of PPs headed by the preposition *with*. Specifically, they develop heuristics for detecting *with* PPs that specify either an `INSTRUMENT` relation or `PART-OF` relation between the attachment site and PP complement. Here, an `INSTRUMENT` PP is one where the complement refers to a tool or implement used to carry out the action referred to by the verb, as in Example (2.3a). A `PART-OF` PP is one where the complement refers to a part of the referent of the noun, as in Example (2.3b). The relation describes "inalienable possession", so for example, in this sense, your nose would be `PART-OF` your face but your eyeglasses would not.

(2.3) *a. I ate fish with a fork.*
b. I ate fish with bones.

Their heuristics exploit the rather systematic use of certain word patterns to express certain semantic relations in dictionary definitions. For the relations under consideration, some example patterns are:

`INSTRUMENT:`	for, used for, used to, a means for
`PART-OF:`	part of, arises from, end of, member of

Using a common dictionary (Webster's Seventh New Collegiate Dictionary), the heuristics look for such patterns in the definitions of the verb and noun candidate attachment sites and the prepositional complement. Additionally, where an exact match is not found, relation arguments found through these patterns can be linked by following hierarchical chains of definitions. Thus, when seeking to disambiguate Example (2.3b), for example, given the definitions

bone: rigid connective tissue that makes up the skeleton of vertebrates,

fish: any of various mostly cold-blooded aquatic vertebrates . . . ,

the prepositional complement *bones* shares a `PART-OF` relation with *vertebrates*, and the potential noun attachment site *fish* is a direct hyponym of *vertebrates*. Therefore the `PART-OF` relation is very likely to hold between *bones* and *fish*.

It seems clear that Jensen and Binot are providing proof of concept that dictionaries can be valuable resources in language processing, and ambiguity resolution in particular, rather than proposing a complete and competitive PP attachment solution. Some of the limitations to expanding this approach to a more complete range of PP attachment cases are worth noting. Usage of prepositional phrases does not generally map so neatly into semantic relations. Further, there are many ambiguity cases where knowing the relations involved is of little use, or where relations expressed through PPs are not inherent relations that could be extracted from a dictionary, or any other resource. Consider again the following ambiguous sentence:

(2.4) I saw the man in the park.

Whether the act of seeing occurred in the park, or say, from across the street, the PP specifies a locative relation between *park* and either *saw* or *man*—knowing the relation does not help in determining the correct attachment. Also, unlike the inherent relation between *forks* and *eating* or *bones* and *fish*, men are not inherently located in parks nor does the act of seeing inherently occur in parks. As such the relation would not be discoverable in dictionary definitions. Still, their approach is effective for the cases they present, as well as other cases that we will examine below, and the line of reasoning is worth following.

There has been much improvement in terms of the accessibility of lexical and semantic resources for machines since Jensen and Binot's pilot study. Most everything is available online now, and even resources designed primarily for human consumption generally include interfaces for programmatic access. WordNet (Miller, 1995) is a lexical database that has become an invaluable resource across the spectrum of NLP efforts, from sentiment analysis (Andreevskaia, 2009) to query expansion for information retrieval (Voorhees, 1994). In addition to providing glosses for each word entry, it also groups words into sets of synonyms, or *synsets*, which are further linked together through conceptual-semantic and lexical relations. These links provide an explicit hierarchy of terms and concepts not unlike the partial hierarchies Jensen and Binot extract from the dictionary to link terms. Much work has been spent developing metrics to use these links for measuring the similarity and relatedness of terms and concepts [see (Pedersen, Patwardhan, and Michelizzi, 2004)].

WordNet has been successfully employed toward improving PP attachment in a number of cases. Brill and Resnik (1994) use WordNet's concept hierarchy to group terms into conceptual classes, reducing sparsity. Their approach induces transformation rules from a corpus of quadruples similar to the RRR corpus. Each rule applies a transformation—i.e. changes the attachment decision from one candidate to another—based on one or more of the lexemes in the quadruple. For example, one learned rule changes attachment from n to v if the verb candidate is *buy* and the preposition is *for.* The authors experiment with allowing the conditions of these rules to include membership to a WordNet synset, resulting in rules such as: change attachment from n to v if the prepositional complement belongs to the WordNet synset *"time"*, or change attachment from n to v if the noun candidate is a member of the synset *"measure, quantity, amount"*. They observe a marked improvement in attachment accuracy using this word class information over basing all rules solely on exact lexeme matches.

Stetina and Nagao (1997) propose an inspired approach that relies heavily on WordNet. It is noteworthy both for its conceptual elegance and for the fact that after a decade and a half it is still one of the top-performing attachment techniques evaluated on the RRR corpus. Their approach is based on inducing decision trees using a variation of the ID3 algorithm (Quinlan, 1986). Each node in the tree partitions the training set based on either the verb attachment candidate, the noun attachment candidate, or the prepositional complement, depending on which attribute results in the most homogeneous partitioning. (Prepositions are not used as a partitioning attribute as entirely separate decision trees are generated for each preposition.) Crucially, the training set at each node is not partitioned on the actual value of the attribute—the lexeme v, n, or n_c—but on its membership within WordNet synsets. Further, the expansion of the decision tree is interleaved with expansion of the WordNet hierarchy in such a way that increasingly specific synsets are used toward the leaf nodes. A full overview of the decision tree induction procedure is given in Algorithm 2.2.

This approach takes the notion of addressing sparsity through generalizing conceptually similar terms and pushes it to extremes. This generalization is balanced by considering synset membership throughout the WordNet hierarchy, automatically selecting the most appropriate level of generalization or specificity supported by the training data. Still, another important consideration is required to make this level of generalization practical. Recall our first example of linguistic ambiguity, Example (1.1) on page 1. This example highlights the fact that the word *arms* has multiple senses: one in which it refers to weaponry and one in which it refers to body parts. The former sense might be involved in PP attachments such as *kill with small arms*, *destroy with nuclear arms*, *battleship with anti-submarine arms*, while the latter sense might be more likely in attachment such as *chimpanzees with hairy arms*, *bodybuilders with muscular arms.* The inability to distinguish between these two senses may be a source of noise when attaching PPs using lexical co-occurrence statistics, but this noise can be greatly amplified when making conceptual generalizations. These two senses of *arms* belong to entirely different synset hierarchies. Making generalizations using the wrong synset hierarchy can lead to unlikely attachment decisions like *chimpanzees with anti-aircraft missiles* or *kill with hairy legs.* To protect against such errors, Stetina and Nagao apply unsupervised word-sense disambiguation to all quadruple terms prior to inducing attachment decision trees.

2.2.2 Distributional Similarity

Statistical semantics can be applied in place of manually built lexical resources to compare new attachment ambiguities with similar training instances. Here, instead of manually compiled lexicons, dictionaries, and semantic networks serving as the basis for a word's

Algorithm 2.2 Stetina & Nagao's decision tree induction algorithm

```
$T \leftarrow$ set of training quadruples
$A \leftarrow$ {verb, noun, p-complement}
$w_{verb} \leftarrow$ top root synset of WordNet verb hierarchy
$w_{noun} \leftarrow$ top root synset of WordNet noun hierarchy
$w_{p-complement} \leftarrow$ top root synset of WordNet noun hierarchy
procedure INDUCE-TREE($T, A, w_{verb}, w_{noun}, w_{p-complement}$)
    if IS-HOMOGENEOUS(T) then
        return tree leaf with the attachment type of the instances in $T$
    else
        if $A = \emptyset$ then
            $A \leftarrow$ {verb, noun, p-complement}
        end if
        $a \leftarrow x \in A$ resulting in the most homogeneous partition of $T$
        $S \leftarrow$ new sub-tree rooted at $a$
        for all $w_{sub} \in \{x | \text{IS-DIRECT-DESCENDANT-SYNSET}(x, w_a)\}$ do
            if $a$ =verb then
                $P \leftarrow \{(v, n, p, n_c) | (v, n, p, n_c) \in T \wedge \text{IS-HYPONYM}(v, w_{sub})\}$
                $S_{sub} \leftarrow$ INDUCE-TREE( $P, A - \{a\}, w_{sub}, w_{noun}, w_{p-complement}$ )
            else if $a$ =noun then
                $P \leftarrow \{(v, n, p, n_c) | (v, n, p, n_c) \in T \wedge \text{IS-HYPONYM}(n, w_{sub})\}$
                $S_{sub} \leftarrow$ INDUCE-TREE( $P, A - \{a\}, w_{verb}, w_{sub}, w_{p-complement}$ )
            else if $a$ =p-complement then
                $P \leftarrow \{(v, n, p, n_c) | (v, n, p, n_c) \in T \wedge \text{IS-HYPONYM}(n_c, w_{sub})\}$
                $S_{sub} \leftarrow$ INDUCE-TREE( $P, A - \{a\}, w_{verb}, w_{noun}, w_{sub}$ )
            end if
            link sub-tree $S_{sub}$ as child of $S$
        end for
        return $S$
    end if
end procedure
```

meaning and relation to other words, the context in which a word occurs serves as its meaning. That is, we can garner some notion of the meaning of *book* by observing that it frequently co-occurs in the same context as words like *read*, *write*, *text*, and *publish*. This view of semantics is based on the distributional hypothesis (Harris, 1985; Firth, 1968), which posits that words with similar meanings tend to occur in similar contexts. The utility of this notion is evinced by its application in a wide array of NLP areas including word-sense disambiguation (Dagan, Lee, and Pereira, 1997), inference in question answering (Lin and Pantel, 2001), and automated building of semantic taxonomies (Snow, Jurafsky, and Ng, 2005), to name just a few.

The context of a word can be defined in various ways. One popular formulation is based purely on proximity. Here the context of a word is defined as the words directly preceding and following it, within a window of n words, where n is a parameter that can be optimized for the task at hand. A vector representing the meaning of a word can then be constructed by observing all occurrences of that focus word in a large corpus and counting the frequencies of co-occurring context words. The dimensionality of the vector can be variable, if the frequency of every context word encountered for the given focus word is included, or fixed to include only a predefined set of context words, such as the C most frequent words across the corpus or some set of semantic primitives. The position of context words can be taken into account (say counting occurrences before the focus word separately from those that occur after it) or context can be treated as a bag-of-words where the position of context words is ignored.

However context is defined and vectors constructed, the power of this approach is that it provides a semantics that can be computed automatically without laborious human intervention. Further, co-occurrence vectors representing words can then be manipulated using vector arithmetic, allowing comparisons to be made between words without reference to manually defined relations. For the task of PP attachment, each word in attachment tuples can be represented by a co-occurrence vector, allowing any of several vector similarity metrics to be applied to assess the similarity of an ambiguous attachment with training instances for which the correct attachment is known. We devote the remainder of this section to the review of two PP attachment approaches that apply this strategy using the k-nearest neighbor algorithm.

Zavrel, Daelemans, and Veenstra (1997) use proximity-based context in their construction of semantic vectors. They define their context window as the two words to the left and two words to the right of a focus word, but consider only words from a fixed set of context words—the 250 most frequent words in their corpus. The position of context words is held as relevant and separate frequency counts are maintained for each of the four context positions, yielding 1000-dimensional vectors. These are subjected to principal component analysis to reduce their dimensionality.

In order to disambiguate PP attachments, these semantic vectors are used instead of the head words in attachment tuples. The form of an attachment tuple is then $(\mathbf{v}, \mathbf{n}, \mathbf{p}, \mathbf{n_c})$, and the distance between two tuples is the sum of the vector distances between each of their components, that is

$$d[(\mathbf{v}, \mathbf{n}, \mathbf{p}, \mathbf{n_c}), (\mathbf{v}', \mathbf{n}', \mathbf{p}', \mathbf{n_c}')] = d(\mathbf{v}, \mathbf{v}') + d(\mathbf{n}, \mathbf{n}') + d(\mathbf{p}, \mathbf{p}') + d(\mathbf{n_c}, \mathbf{n_c}').$$

Distances between vectors are calculated using a variation of cosine similarity. An ambiguous attachment converted into vector form is thus compared to training instances, also in vector form, and the attachment is decided according to the known attachments of the k nearest training instances, each of which is considered with a weight proportional to its distance.

Zhao and Lin (2004) use dependency relations instead of token proximity as their basis for a word's context. Dependency relations are a form of syntactic analysis different from the phrase structure analysis introduced in Chapter 1. Phrase structure analysis is concerned primarily with constituency—or how phrases combine to make other phrases. In contrast, dependency graphs like the one in Figure 2.2, relate heads to dependent words, or modifiers. A full comparison of the strengths and weaknesses of dependency parsing and phrase structure parsing is beyond the scope of our discussion. However, dependency relations do offer a more direct representation of the relationships between words, which may be more appropriate here for determining the context of a word.

Given a dependency graph for the sentence in which a focus word occurs, the context of the focus word is then defined as the set of all words with which it shares a dependency relation, along with the type of that dependency. For instance, the context of *solution* in

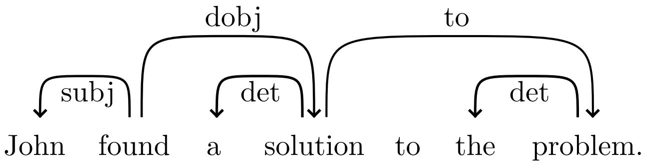


Figure 2.2: An example dependency graph: subject, direct object, and determiner relation types are abbreviated as subj, dobj, and det, respectively

the dependency graph of Figure 2.2 would be the dependency-type/word pairs

$$(det, a), (to, problem), (-dobj, found),$$

where $-dobj$ indicates the inverse relation of $dobj$.

While syntax gives a much more precise view of a word's context, relying on syntactic analysis has its disadvantages. The main appeal of a statistical semantics is that it should not require painstaking encoding of knowledge by humans. Thus, this approach would be of little value if it required manually annotated dependency graphs, yet automatically generated dependency graphs are subject to the very kind of errors that we hope to reduce through improved PP attachment. In fact, in our example above, extracting the context of *solution* relies in part on correctly determining that the ambiguous PP *to the problem* complements *solution* and not *found* when building the dependency graph.

As in the previous approach, ambiguous attachments are decided based on the weighted votes of the k most similar training instances. Depending on the situation, one of several increasingly less restrictive definitions of similarity and the corresponding set of nearest neighbors is applied, in a fashion resembling backing off. Again, backing off allows the use of the most detailed model when sufficient information is available, and a coarser model of similarity when sparsity prevents this. Each successive definition is applied only if the previous one is unable to make an attachment decision, and a default of noun attachment applies if the last definition fails. They are defined as follows:

1. The nearest neighbors are tuples that match exactly the input tuple. Similarity between each of these is 1.

2. The nearest neighbors are the k most similar tuples that have the same preposition as the input tuple. Similarity between two tuples is defined as

$$sim[(\mathbf{v}, \mathbf{n}, \mathbf{p}, \mathbf{n_c}), (\mathbf{v}', \mathbf{n}', \mathbf{p}', \mathbf{n_c}')] = ab + bc + ca$$

   where a, b, c give a measure of the distributional similarity between $\mathbf{v}$ and $\mathbf{v}'$, $\mathbf{n}$ and $\mathbf{n}'$, and $\mathbf{n_c}$ and $\mathbf{n_c}$‘, respectively.

3. The nearest neighbors are the k most similar tuples that have the same preposition as the input tuple. Similarity between two tuples is defined as

$$sim[(\mathbf{v}, \mathbf{n}, \mathbf{p}, \mathbf{n_c}), (\mathbf{v}', \mathbf{n}', \mathbf{p}', \mathbf{n_c}')] = a + b + c$$

4. The nearest neighbors are all tuples that have the same preposition as the input tuple. Similarity between each of these is constant—i.e. the attachment of each nearest neighbor is weighted equally.

2.3 Unsupervised Attachment

We have established that PPs and their attachment can be a significant source of ambiguity in text. However, PPs need not always occur in ambiguous contexts. Consider for example the following sentence:

> *(2.5) The man in the park looked through the telescope.*

Here, there is no question as to where either of the two PPs attach. The PP *in the park* must modify the noun *man*, and *through the telescope* must complement the verb *looked*—no alternative attachment sites are available for either PP. Returning to our formalization of PP occurrences as tuples, unambiguous PPs like these yield triples of the form (n, p, n_c) and (v, p, n_c), where the attachment site is known to be n and v, respectively. The triples from this particular sentence would then be $(man, in, park)$ and $(looked, through, telescope)$.

Unlike the methods discussed thus far, no manually annotated data are required to extract such triples and compile statistics over them. Accordingly, methods that use such information are called unsupervised methods. To be clear, these are not unsupervised in the sense that they attempt to highlight hidden structure in the data or cluster similar instances, as is conventionally what is meant by unsupervised learning. The aim is to build an attachment prediction model much like those built by the supervised methods described above. In a sense, these methods can be thought of as author-supervised instead of annotator-supervised: the correct answer to each attachment training instance is labeled implicitly in the author's use of unambiguous language.

There are many patterns in which PPs occur unambiguously, Example (2.5) merely illustrates the simplest and easiest to identify even without a preliminary parse of the sentence. Indeed, unambiguous attachment patterns can be detected with simple string matching, or with increasing flexibility using part-of-speech tags, phrase chunks, or preliminary parses. An early approach to unsupervised attachment (Ratnaparkhi, 1998) applies heuristics to detect unambiguous attachments. Unlabeled text is first tagged for part of speech, then simple noun phrases and quantifier phrases are chunked and replaced with their head words. Finally, verb attachment triples, (v, p, n_c), are extracted in instances where

- v is the first verb that occurs within K words to the left of p,
- v is not a form of the verb *to be*,
- no noun occurs between v and p,
- n_c is the first noun that occurs within K words to the right of p,
- no verb occurs between p and n_c,

and noun attachment triples, (n, p, n_c), are extracted where

- n is the first noun that occurs within K words to the left of p,
- no verb occurs within K words to the left of p,
- n_c is the first noun that occurs within K words to the right of p,
- no verb occurs between p and n_c.

Note that these rules do allow for extraction of incorrect triples. Even with perfect part-of-speech tagging and chunking, which is impossible to guarantee on all texts, the resulting triples would likely contain errors. In particular, the noun attachment heuristic accepts as unambiguous any cases with multiple noun attachment sites or even canonically ambiguous cases for some values of K, as in the following example:

(2.6) a. *John gave the book about syntax to Mary.* $\Rightarrow *(syntax, to, Mary)$
b. *John gave Mary the book for her birthday.* $\Rightarrow *(book, for, birthday)$

The verb attachment heuristic seems sound in comparison but it too accepts incorrect triples in cases involving non-canonical ambiguity. Specifically, it accepts as unambiguous any cases where v is not the main verb but part of a reduced relative clause as in:

(2.7) a. *Jill bought the house Jack built with his own hands.* $\Rightarrow (built, with, hands)$
b. *Jill bought the house Jack built with her own money.* $\Rightarrow *(built, with, money)$

The impact of such errors, and any others, can be measured by performing the unsupervised extraction over annotated texts so that the resulting triples can be compared against the actual attachments. Ratnaparkhi reports that only 69% of supposedly unambiguous triples extracted from annotated PTB data have the correct attachment. His take is that the noisiness of the resulting triples is offset to some degree by their abundance.

More recent approaches (Volk, 2001; Olteanu and Moldovan, 2005) leverage the enormous size of the Web and the power of web search engines to gather statistics on unambiguous PP attachments. Volk (2001) uses AltaVista,[2] and its `NEAR` operator, which requires its arguments to occur within 10 words of each other. The frequency of occurrence of an unambiguous verb attachment triple, $f(v, p, n_c)$, is approximated by the number of documents returned for the query v `NEAR` p `NEAR` n_c. Similarly, $f(n, p, n_c)$ is approximated by the number of hits for the query n `NEAR` p `NEAR` n_c. Olteanu and Moldovan (2005) use Google,[3] and its exact phrase search (quoted) and wildcard operators. Here, $f(v, p, n_c)$ is approximated by the number of hits for the queries "$v\ p\ n_c$" and "$v\ p\ *\ n_c$", and $f(n, p, n_c)$ by "$n\ p\ n_c$" and "$n\ p\ *\ n_c$". These Web-based approaches are susceptible to the same noise as Ratnaparkhi's heuristic approach. However, the justification of quantity over quality is even more apt here, as the Web is several orders of magnitude larger.

Yet another approach challenges this notion of the quantity of extracted instances justifying their low quality. Kawahara and Kurohashi (2005) use heuristics over tagged and chunked text in a similar way as Ratnaparkhi. However, their heuristics are much more restrictive. Unambiguous noun attachment triples are extracted only where an NP at the beginning of a sentence is directly followed by a PP, as is the case for the first PP in Example (2.5). Barring tagging or chunking errors, the resulting noun triples are all truly unambiguous. Unlike Ratnaparkhi's noun triple heuristic, this one correctly ignores PPs like those in Example (2.6).

Their verb triple heuristic does not offer as much of an improvement. It exploits the fact that pronouns are not viable attachment candidates. Accordingly, it extracts verb triples wherever a verb, a pronoun, and a PP occur in direct succession as in Example (2.8) below:

(2.8) She $[_{verb}$ *sent*$]$ $[_{pronoun}$ *him*$]$ $[_{PP}$ *into the nursery*$]$ *to gather up his toys.*

[2] `http://www.altavista.com`
[3] `http://www.google.com`

While this should be slightly more accurate than Ratnaparkhi's verb triple heuristic, it is still susceptible to errors involving multiple verb candidates. Consider the following modification of Example (2.7):

(2.9) *a. Jill loved the house Jack built her with his own hands.* $\Rightarrow (built, with, hands)$
b. Jill loved the house Jack built her with all her heart. $\Rightarrow *(built, with, heart)$

Additionally, it is not clear why they omit cases where a PP directly follows an intransitive verb like the last PP in Example (2.5). These are just as unambiguous, and just as easy to detect as verbs with pronominal direct objects, but far more frequent.

Overall, Kawahara and Kurohashi compensate for the restrictiveness of the heuristics by using a much larger unlabeled corpus, compiled from 200 million web pages containing 1.3 billion sentences. In contrast, Ratnaparkhi used 970 thousand unlabeled sentences, yielding 910 thousand supposedly unambiguous triples, almost one triple for every sentence. Kawahara and Kurohashi extract 168 million triples from their unlabeled corpus. Thus even with a much lower yield per sentence, they are able to compile a model with more instances by orders of magnitude. Unfortunately, they provide no direct evaluation of the accuracy of their extracted triples.

Regardless of the methods for detecting and extracting them, such unambiguous occurrences can be exploited to compile statistics from large collections of entirely unlabeled text, avoiding the considerable cost of manually annotating corpora for new domains or languages, or for supplementing existing resources. When faced with an ambiguous occurrence, (v, n, p, n_c), we can compare the frequency of unambiguous occurrences where the PP in question attaches to each of the given attachment candidates. As we have seen in Collins and Brooks' backed-off model, triples can be almost as effective at deciding attachment as full quadruples. However, the triples obtained from unambiguous occurrences cannot provide counter examples [e.g. (V, n, p, n_c)], only positive examples, nor can they inform on the co-occurrences between verb and noun attachment sites [e.g. (v, n, p)]. As with the aforementioned noise issues, these shortcomings are mitigated—it is hoped—by the sheer quantity of data that is available at negligible cost compared to annotated data.

Chapter 3

Beyond Toy Evaluations

Our survey of the field in the previous chapter has omitted any quantitative assessment or comparison of the various techniques described therein. Yet, as with any scientific inquiry or engineering endeavor, some way of measuring success is necessary to predict how well a model will perform in reality and to point future efforts in the most promising direction.

In this chapter, we look at the problem of evaluating PP attachment techniques realistically and meaningfully. In particular, we identify three dimensions on which to consider evaluation: task formulation, baseline comparison, and input realism, focusing on the latter two in this chapter. We have already described the traditional task formulation (binarity, head dependency, independence) in Section 1.3, and defer discussion of more complete formulations and related evaluation concerns to Chapters 4 and 5.

3.1 Baselines

In natural language processing, as in many other disciplines, evaluation is essential in that it affords some notion of progress. Distinguishing progress from regression allows us to select superior models, features, and parameters, and to point efforts in the right direction. However, the ability to quantify performance and to assign "better" numbers to better systems does not give a complete picture. It is easy enough to see that a system that predicts, say, the outcomes of coin tosses with 30% accuracy is better than one that predicts outcomes with 20% accuracy. Without some notion of a coin-toss-prediction baseline, though, the fact that both systems perform much worse than the simplest predictor (chance) may not be obvious.

When first starting to tackle a new problem, the only option for a baseline may be to evaluate a naive, or trivially implementable, technique. For example, in the case of normally distributed binary decisions, like coin tosses, randomly guessing the answer should yield the correct response approximately 50% of the time. For PP attachment, the structural principles of minimal attachment or right association are easily implemented to provide a preliminary baseline. A better baseline can be formulated by taking into account the importance of the preposition in determining an attachment decision (Collins and Brooks, 1995), and for a given PP, assigning the attachment (verb or noun) most commonly found for PPs with the same preposition in a training set. The accuracy of these baselines on the RRR corpus (Ratnaparkhi, Reynar, and Roukos, 1994) is given in Table 3.1, along with the accuracy of a sampling of the techniques described in the previous chapter.

While naive baselines provide an essential preliminary point of comparison for the first

Table 3.1: Accuracy of various attachment methods versus naive baselines on RRR corpus

	PP attachment accuracy (%)
Minimal attachment	41.0
Right association	59.0
Majority by preposition	72.2
Maximum entropy (Ratnaparkhi, Reynar, and Roukos, 1994)	81.6
Backed-off maximum likelihood estimation (Collins and Brooks, 1995)	84.5
Memory-based learning (Zavrel, Daelemans, and Veenstra, 1997)	84.4
Decision trees with word-sense disambiguation (Stetina and Nagao, 1997)	88.1

attempts at a new problem, as the state of the art progresses, so too should the baseline. It would hardly be noteworthy today to offer an attachment technique yielding a few percentage points improvement over right association when over a decade and a half ago state-of-the-art attachment techniques were capable of 20-30% better accuracy. Unfortunately, evaluations reported in the literature may not always be directly comparable due to differing data sources, task formulations, evaluation paradigms, etc., and re-implementing or adapting the state of the art for the sake of comparison is rarely anywhere near as trivial as implementing naive baselines. Consequently, shared tasks, resources, and evaluations, which provide a uniform yardstick for all, are indispensable in many NLP communities.

For PP attachment, that yardstick has generally been the RRR corpus. It has undoubtedly been an invaluable resource for the development of PP attachment techniques, but there are several reasons why it can no longer serve as the de facto standard data set against which all baseline evaluations are computed. To start, it is only applicable to the canonical form of PP attachment. A large part of the utility of the RRR corpus is that it offers a simplified, abstract view of a rather complex problem. It is not a corpus of natural language text per se, but rather a collection of filtered, preprocessed data derived from natural language text in order to distill the essentials of one particular view of the attachment problem. As such, it cannot support inquiry beyond this view, the need for which is a central premise of this thesis. The non-canonical ambiguity cases that need to be addressed and the additional contextual cues necessary for their attachment are simply not a part of the RRR corpus.

More important to the present discussion, comparing the performance of new attachment techniques against most of the classical approaches evaluated on the RRR corpus is no longer particularly indicative of what to expect on any real-world task. In the early days of PP attachment, the real-world attachment task was to take the preliminary syntactic analysis of a partial parser or chunker and produce a more complete analysis by attaching ambiguous PPs, which could not be handled by the parser. Today parsers are generally capable of complete syntactic analysis including PP (and other ambiguous) attachments, thanks in part to the lessons learned from early work in PP attachment. Now, the real-world task for PP attachment is thus to improve the attachment accuracy of an already complete syntactic analysis. Accordingly, while comparing performance against other attachment techniques may be informative in some cases, the new absolute minimum baseline must be the attachment performance of the parser.

Unfortunately, evaluations against a parser's attachment performance are largely absent from the literature, and re-evaluating under more realistic conditions yields surprisingly disappointing results. Among the first to present the need for more realistic PP attachment evaluation, Atterer and Shütze (2007) demonstrate the overly optimistic view given by traditional evaluation methodology. They compare the attachment performance of a state-of-the-art parser (Bikel, 2004) against the backed-off model, as well as two more recent PP attachment approaches (Olteanu and Moldovan, 2005; Toutanova, Manning, and Ng, 2004), showing that none of the three PP-attachment-specific techniques are capable of performing markedly better than the more general parser.

This result delivers a heavy blow to the status quo of PP attachment evaluation, but this is softened somewhat when considering some of the choices Atterer and Schütze make for their comparisons. For one, the comparison of Bikel's parser with Collins and Brooks' backed-off model is not particularly useful. Bikel's parser re-implements Collins' parsing model (Collins, 1999), which explicitly includes the lexical strategies leveraged in his prior PP attachment work. It would thus be rather surprising if there were a significant performance difference between the two. In the case of the approaches of both Olteanu and Moldovan and Toutanova et al., the primary contribution is to incorporate a diverse set of features that combine to provide an attachment advantage. Atterer and Schütze's comparison, however, uses figures that represent baseline versions of these attachment systems that do not use any of the non-canonical features, which are not available in the RRR corpus. The version of Olteanu and Moldovan's system that they compare against, for example, was only meant to be a rough baseline for their experiments with additional features. It uses only lexical quadruples—without even the morphological preprocessing used by the backed-off model. In our own experiments in Chapter 5 based on Olteanu and Moldovan's feature set, we observe a noticeable performance improvement from their additional features, as compared to merely applying a support vector machine to lexical quadruples.

Nonetheless, Atterer and Schütze's fundamental message—the need for more realistic evaluation and an appropriate baseline such as a parser—is fitting. As we will see throughout the rest of this thesis, it is remarkably difficult to provide a significant improvement over the attachment performance of a state-of-the-art parser in a realistic context.

3.2 Input Realism

The quadruples of the RRR corpus are extracted directly from the manual annotations of the Penn Treebank (PTB). Accordingly, the information in each quadruple is perfect (at least in theory)—quadruples are extracted wherever the appropriate ambiguity exists and nowhere else, and the potential attachment sites and all relevant heads are identified without error. Atterer and Schütze describe quadruples derived in this way as provided by an oracle: a source of infallible information which is not present in the data in its naturally occurring form (and which an automated tool could not extract without incurring some degree of error).

Real natural language text does not include preprocessed quadruples representing attachment ambiguities. On a real-world task, a parser must produce a syntactic analysis of a given string of text, an extractor must identify and extract ambiguities from this analysis, and only then can an attacher consider the evidence and select an attachment. The parser will inevitably make mistakes resulting in failure to extract some quadruples or elements thereof. For example, the parser can fail to identify or mis-parse one of the attachment candidates, or fail to recognize the PP at all. If a PP or one (or both) of its potential attachment sites are not identified, no quadruple can be extracted for examination by an attacher. If attachment candidates are identified or extracted incorrectly the attacher may make a decision based on misleading evidence.

As such, the performance of an attacher given input quadruples from an oracle is likely to be higher than that of an attacher given more realistic input from a parser/extractor. A truly realistic evaluation of PP attachment would thus benefit from some idea of the degree of this difference wherever oracles must be used.

3.2.1 Atterer & Shütze's Overstatement of the Oracle Crutch

Atterer and Schütze (2007) offer a comparative evaluation between attachment using oracle-derived RRR quadruples or a parser working on the raw underlying text of these quadruples. They use Bikel's parser (2004) in both cases, evaluating its attachment accuracy on either RRR quadruples[1] or on the original full-text sentences from which RRR quadruples were extracted. They demonstrate that oracle use provides a marked performance advantage, arguing that the use of oracle-derived quadruples as input serves as a crutch to attachers, artificially bolstering their performance. In this view, oracle-based evaluation of attachers "allows no inferences as to their real performance," and sound, realistic evaluation methodology requires doing away with oracles, above all else.

The gravity of their concern hinges on the way they incorporate into their evaluation RRR quadruples that the parser fails to extract from the raw text. We will refer to such occurrences, hereafter, as quadruple extraction failures (QEFs).[2] Atterer and Schütze incorporate quadruple extraction failures into their evaluations in various ways, including ignoring them and evaluating only the quadruples that could actually be extracted (`QEF-discard`), or scoring them as errors (`QEF-error`). Each approach presents a slightly different interpretation of attachment performance, neither of which is ideal.

Discarding quadruple extraction failures effectively ignores most of the additional challenge of using a parser for realistic input, and is qualitatively—and quantitatively, in Atterer and Schütze's and our own experiments described below—similar to traditional, oracle-based evaluation. Failing to recognize an attachment ambiguity when performing real-world language processing results in a less accurate syntactic analysis and degrades understanding of the text. While the blame does not lie with the attacher per se, quadruple extraction failures should be reflected in evaluation of a real-world attachment system.

Counting quadruple extraction failures as errors seems to provide a more realistic assessment until we look at them, and the RRR quadruples, more closely. We define a quadruple extraction failure as a failure to extract an attachment quadruple where one is present in the RRR corpus. Intuitively, this could only indicate an error on the part of the parser or some other preprocessing component. But our use of the RRR corpus here is not consistent

[1] Bikel's parser works over sentences rather than PP attachment quadruples—as is generally the case with all natural language parsers. Thus, to simulate oracle-derived RRR quadruples, Atterer and Schütze construct artificial sentences from each RRR quadruple by prepending the words in the quadruple with the subject NP *They*. For example, $(saw, man, with, telescope)$ yields the sentence "They saw man with telescope." These artificial sentences maintain the relevant qualities of oracle-derived quadruples: the attachment ambiguity and the relevant heads are obvious.

[2] Atterer and Schütze refer to quadruple extraction failures as non-attachment cases.

with its design goals, and in the context of our use, it is not the infallible gold standard that might be expected. The RRR corpus was not designed to faithfully represent the attachment ambiguities of its underlying source text, but rather to generate from it as many correct quadruple instances as possible. Accordingly, many RRR quadruples do not reflect the actual ambiguity present in the original text. While every quadruple in the corpus presents a binary attachment ambiguity between verb and noun attachment, looking at the underlying text of some quadruples reveals PPs with greater ambiguity or none at all. Example (3.1) gives two RRR quadruples, each supposedly representing a canonical binary attachment ambiguity, and the underlying non-binary ambiguities from which they were extracted.

(3.1) a. *Japan, et al.* [$_v$ *provide*] *about* [$_n$ *80%*] *of the* [$_n$ *steel*] [$_v$ *imported*] *to the* [$_n$ *U.S.*] [$_{PP}$ *under the quota program*]
$\Rightarrow (V, provide, \%, under, program)$

b. *the arbitragers started* [$_v$ *dumping*] [$_n$ *positions*] *in their entire* [$_n$ *portfolios*], *including major blue-chip* [$_n$ *stocks*] *that have* [$_v$ *had*] *large price* [$_n$ *runups*] *this* [$_n$ *year*] *,* [$_{PP}$ *in a desperate race*] *for cash*
$\Rightarrow (V, dumping, positions, in, race)$

Such cases are rightly ignored by a parser/extractor looking for binary attachment ambiguities. In effect, the quadruple extraction failures observed by Atterer and Schütze may be just as likely attributable to erroneous quadruples in the RRR corpus as to parser error.

The preliminary nature of the underlying treebank (PTB-0.5) and/or shortcomings in the extraction procedure used to extract quadruples for the RRR corpus result in additional issues. Quadruples are present in the RRR corpus where the underlying text contains no PP at all, as in Example (3.2a) below for example, where the infinitive VP *to shop* is mistakenly extracted as a PP or in Example (3.2b), where part of the quantifier phrase *60 to 100* is mistakenly extracted as a PP.

(3.2) a. *consumers…feel they have less time* [$_{VP}$ *to shop*]
$\Rightarrow (V, have, time, to, shop)$

b. *the market would have to drop* [$_{NP}$ [$_{QP}$ *60 to 100*] *points*]
$\Rightarrow (N, drop, 60, to, points)$

c. *The Senate began debating its \$14.1 billion deficit-reduction bill for fiscal 1990,* [$_{PP}$ *with* [$_S$ *Democratic leaders asserting that the measure will be approved quickly and without a cut in the capital-gains tax*]].
$\Rightarrow (V, debating, bill, with, asserting)$

A particularly disturbing class of quadruples in the RRR corpus is that of quadruples derived from PPs with clausal complements, as in Example (3.2c). Not only do these not fit the canonical form of ambiguity, but they simply cannot be encoded felicitously in the quadruple representation, which defines the last element, n_c, as the noun that heads the NP complement of the PP. The formalism simply has no means of encoding that n_c is actually a verb heading a sentence.

Some of these issues even extend beyond quadruple extraction failures, such that the ambiguity resolution task presented to the parser/extractor/attacher is fundamentally different from that presented by the oracle-derived RRR quadruple. For example, the RRR corpus systematically errs on head extraction of some complex named entities, like *the Unites States Department of Agriculture* or *the Imperial Ballet of St. Petersburg*, where the head is given as the token *the*, as in the following:

(3.3) *she dominated the Imperial Ballet of St. Petersburg through her dancing*
$\Rightarrow (V, dominated, the, through, dancing)$

Ignoring the fact that this again presents non-binary ambiguities as binary ambiguities whenever the named entity includes a PP, these "errors" on the part of the RRR extraction procedures have an effect similar to that of some of the morphological preprocessing techniques employed by Collins and Brooks in their backed-off model. In effect, a large class of attachment candidates are being replaced with a symbol representing that class, thereby reducing sparsity in the same way that Collins and Brooks do by replacing capitalized nouns with the symbol `NAME`.

In short, the RRR corpus does not—nor was it meant to—provide a faithful representation of the text from which it was extracted. Accordingly, RRR quadruples that a parser cannot extract from the underlying text should not necessarily be counted as errors. Summarily treating these quadruple extraction failures as errors leads Atterer and Schütze to overestimate the performance difference between oracle- and parser-derived input. In the remainder of this section we report on our own attempts to improve upon their experiments. First, we investigate the extent of quadruple extraction failures attributable to RRR corpus deficiencies. We then compare oracle versus parser input more accurately using a modern treebank, where such issues do not arise.

Experiment 1: Examining Quadruple Extraction Failures

To get a more accurate assessment of the impact of oracle input on attacher performance, we perform a similar experiment to that of Atterer and Schütze with additional analysis taking into account the nature of the RRR source-text-to-quadruple extraction. Specifically, rather than considering all quadruple extraction failures either as errors (`QEF-error`) or as not part of the evaluation (`QEF-discard`), we manually evaluate each case individually to ascertain the reason why the attachment ambiguity was not recognized (`QEF-evaluate`). Where the quadruple extraction failure is the result of a parser error, it is included in our evaluation (as it would be with `QEF-error`), but we do not penalize for quadruple extraction failures where they correctly represent the underlying text.

We use our own re-implementation of Collins and Brook's backed-off model, assessing its performance on both oracle-derived input and parser-derived input. We use Charniak's parser (Charniak and Johnson, 2005) to generate parses from raw text, and extract quadruples from these parses (see Appendix A for details on identifying attachment candidates and extracting heads).

Our test data was generated by parsing the underlying sentences of each of the quadruples in the RRR test set.[3] The correct attachment for each quadruple that could be extracted from the automatically generated parses was determined from the corresponding RRR quadruple. As did Atterer and Schütze, we use the latest version of the Penn Treebank (PTB-3) to generate our training data, specifically the standard sections 2-21 of the WSJ corpus. (We verified that there is no overlap between these sections and the RRR test set.) Again, the underlying raw text was parsed using Charniak's parser and quadruples were extracted for all canonical attachment ambiguities. In all, roughly 33 000 quadruples were extracted for training. The correct attachments for each quadruple were then determined from the treebank annotations.

[3] We were unable to find the underlying text for eight of the quadruples in the RRR test set.

Of course, not all of the RRR test quadruples could be generated from the automated parses of the underlying text, and the main goal of the experiment was to examine these cases. A total of 238 quadruple extraction failures were observed. For each of these, we manually analyzed the original RRR quadruple, the underlying text, the corresponding PTB-0.5 annotations, and whenever possible the corresponding PTB-3 annotations. Based on this analysis, each quadruple extraction failure was classified as resulting from parser error, annotation error, or non-canonical ambiguity, which we define as follows:

parser error: The syntactic analysis provided by the parser is incorrect in a way that precludes extraction of a corresponding quadruple. Such errors include failure to recognize the prepositional phrase or failure to recognize two attachment candidates. Note that finding the wrong attachment candidates does not necessarily warrant inclusion in this class, as long as exactly two candidates are found.

annotation error: The original quadruple from the RRR corpus indicates an attachment ambiguity that is not actually present in the text. These generally involve constituents that are erroneously annotated as PPs in the preliminary PTB-0.5, such as the adverb phrase *at all* or a numerical range specified in a quantifier phrase, as in Example (3.2a). Also included here are unambiguous PPs that are part of ambiguous ADVPs, such as *to development* in “*banks aim to boost their U.K. presence* [$_{ADVP}$ *prior* [$_{PP}$ *to development*]].*”* While these annotation errors are more or less obvious, our classifications here are not based on intuitive judgments, but rather the latest annotation guidelines of the Penn Treebank, and wherever possible, the corresponding PTB-3 annotations.

non-canonical: The original quadruple from the RRR corpus indicates an attachment ambiguity that corresponds to a non-canonical attachment ambiguity in the actual text. These include PPs with multiple noun attachment candidates, multiple verb attachment candidates, adjectival and adverbial attachment candidates, etc., as exemplified in Example (3.1).

The distribution of quadruple extraction failures that we observed using this classification scheme is depicted in Figure 3.1, from which it is plainly visible that parser error accounts for only a small minority of the occurrences.

Table 3.2 summarizes the comparative performance of the backed-off model when given oracle-derived or parser-derived input from the RRR corpus. Since some quadruples are discarded for some treatments of quadruple extraction failures, the number of quadruples included for each evaluation is also shown. Our manual analysis of quadruple extraction failures is factored into the `QEF-evaluate` measure, where parser-error QEFs are counted as attachment errors and all other QEFs are discarded. While there is a noticeable decrease in performance with parser input compared to oracle input, the effect is much less severe than is suggested when all QEFs are counted as errors (`QEF-error`). Whether this smaller difference in performance entirely mitigates Atterer and Schütze’s view of oracles as the key impediment to realistic evaluation is subject to interpretation. However, it does certainly demonstrate how much of an impact the shortcomings of the RRR corpus can have on realistic evaluation.

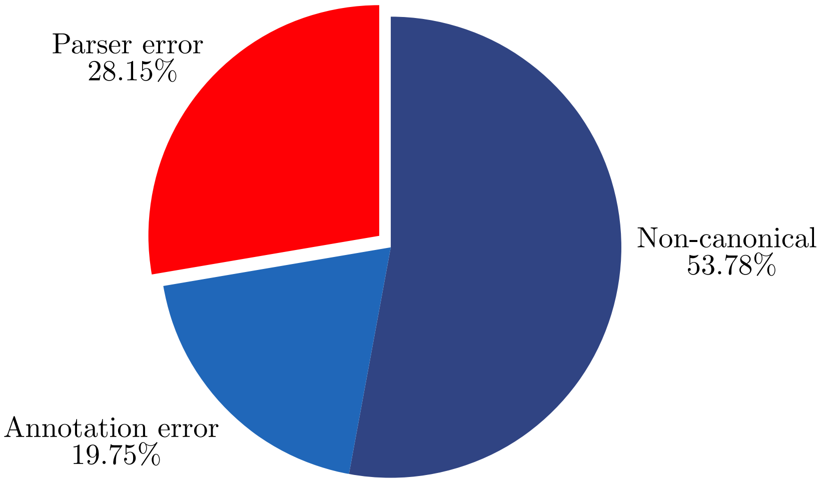


Figure 3.1: Distribution of quadruple extraction failures

Table 3.2: Backed-off model performance on oracle versus parser input on the RRR corpus

Evaluation type		Test size	PP attachment accuracy (%)
Oracle input		3097	84.30
Parser input	(QEF-discard)	2847	84.37
Parser input	(QEF-error)	3092	77.62
Parser input	**(QEF-evaluate)**	**2958**	**81.14**

QEF-evaluate gives a better sense of the difference between attachment using oracle or parser input, but it is still not entirely accurate. We have accounted for quadruple extraction failures that are the result not of parser performance issues, but of changing annotation standards and peculiarities of RRR extraction methods (where fidelity in representing the underlying text is not a concern). These issues are not limited to quadruple extraction failures, thus even with our improved accounting of QEF cases the comparison here between oracle and parser input is a forced one; the alternatives are not being compared on equal footing.

Experiment 2: True Comparison of Oracle versus Parser Input

A true comparison of the effect of oracle versus parser input on PP attachment requires that the data used for both alternatives, for both training and testing, come from the same underlying text, where manual annotations follow the same annotation standard and quadruples are extracted in the same manner. To that end, we again compare the performance of the backed-off model using oracle- and parser-derived quadruples, this time using a more appropriate data set: the WSJ corpus of PTB-3.

For this experiment, our data was generated from the WSJ corpus of PTB-3 for both oracle and parser input. We use sections 2-21 for training and section 23 for testing. For our oracle data sets, quadruples are extracted directly form the treebank annotations (see Appendix A for details). For our parser data sets, the raw text from the corpus was parsed using Charniak's parser and quadruples were extracted from these parses using the same techniques. A total of about 33 000 training quadruples were extracted for each of the oracle and parser data sets. In both cases, the correct attachment of each quadruple was determined from the treebank annotations.

Because PTB-3 is not impaired by the shortcomings and inconsistencies of the preliminary version 0.5, we are able to extract our oracle-derived input directly from the treebank. Accordingly, we are not encumbered by the issues demonstrated in our previous experiment. Specifically,

- the treebank annotations are consistent and adhere to sound guidelines, making annotation-error QEF cases virtually impossible;
- we can ensure that all quadruples accurately represent the ambiguity of the underlying text, and that quadruples are only extracted for the canonical form under consideration, thereby eliminating non-canonical QEF cases;
- the same extraction procedure can be used for both oracle and parser experiments, ensuring that any extractor-caused biases do not unfairly penalize one or the other result;[4] and
- the same source text can be used to generate training for both experiments, ensuring that the results are not affected by any differences in training data.

Accordingly, a realistic and fair comparison between oracle and parser input is possible, and we need not even manually verify quadruple extraction failures.

Quadruple extraction failures do still need to be accounted for, and simply counting them all as errors may be suboptimal. The annotations of PTB-3 are well thought-out and consistent. Our parse-to-quadruple extraction procedures, while not perfect, faithfully represent the underlying text, and we apply them consistently to both treebank- and parser-derived trees. As such, there is no doubt that all oracle-derived quadruples are correct and that quadruple extraction failures should therefore be included in our evaluation. But are quadruple extraction failures, even here, necessarily errors? Bear in mind that modern parsers make their own attachment decisions, as we brought up in the previous section in arguing for better attachment evaluation baselines. A quadruple extraction failure can occur where the parser and/or extractor fail to recognize a canonical attachment ambiguity, but the PP may still have been recognized and some attachment decision made. Because we have access to the underlying treebank annotations in this experiment, we can evaluate the parser's attachment decision in these cases where no quadruple is extracted. Thus, in this experiment, we introduce another way of handling quadruple extraction failures in our evaluation: `QEF-parser-evaluate`. With this approach, QEF cases are always included

[4] Some concern may arise here that using our automated extractor in generating both oracle and parser quadruples means that our evaluation can no longer fairly account for any shortcomings on the part of the extractor. First, we should note that this is not really different form any experiments using RRR data, including our previous one. The RRR quadruples are not manually extracted from treebank annotations and are thus subject to errors, some of which we have seen in our analysis of quadruple extraction failures. However, any canonical ambiguities that RRR extraction procedures failed to extract are, by definition, not included in the RRR corpus and cannot be factored into any evaluation. Evaluating quadruple extraction against the extracted quadruples of the RRR corpus is therefore rather arbitrary and one-sided: we can observe errors made by our extractor that were handled correctly in RRR, but we cannot observe errors made by the RRR extractor that were handled correctly by our extractor or even cases where both extractors fail to extract a relevant quadruple.

in our evaluation, but the parser's attachment decision is evaluated to assess whether it should be counted as correct (where the parser correctly attaches the PP despite mistaking its degree of ambiguity) or as an error (where the parser incorrectly attaches the PP or fails to recognize the PP at all).

We contend that `QEF-parser-evaluate` provides a more realistic treatment of quadruple extraction failures. The fundamental motivation behind this entire line of experimentation is that oracle-derived quadruples lead to evaluations that do not reflect a realistic attachment task, where preprocessing from raw text to quadruple adds to the challenge. We established that quadruple extraction failures should not be ignored, since they can result in attachment errors, even if these are not directly attributable to an attachment decision module. By the same reasoning, correct decisions that are not directly caused by an attachment module should be given credit. Ultimately, the real-world task of an attacher is to take a parser's syntactic analysis and improve attachments wherever possible. A fair and realistic evaluation should consider all correct and erroneous attachments arising from that process, regardless of which stage of the process is responsible for the final decision.

The accuracy of our implementation of the backed-off model on the WSJ corpus of PTB-3 is given in Table 3.3 for both oracle- and parser-derived input. While we consider `QEF-parser-evaluate` to be most representative of real-world performance, figures for `QEF-discard` and `QEF-error` are also given for the sake of completeness. Compared with the results observed on the RRR corpus when properly accounting for QEF cases (`QEF-evaluate` in Table 3.2), the difference between oracle and parser input is much less pronounced; compared to those observed when considering all QEF cases as errors (`QEF-error` in Table 3.2), the difference is practically negligible. In fact, the difference between the figure for `QEF-parser-evaluate` and that for oracle input is not statistically significant. It hardly warrants Atterer and Schütze's claim that the use of an oracle to evaluate attachers "allows no inferences as to their real performance." Given the results of both this and the previous experiment, it seems considerably more likely that the large performance gap they observe between oracle-based and parser-based input is more a function of the inadequacies of the RRR corpus than of any fundamental principle.

Table 3.3: Backed-off model performance on oracle versus parser input on the WSJ corpus

Evaluation type		Test size	PP attachment accuracy (%)
Oracle input		1760	86.14
Parser input	(`QEF-discard`)	1615	86.75
Parser input	(`QEF-error`)	1760	79.20
Parser input	**(`QEF-parser-evaluate`)**	**1760**	**84.26**

This is not to say that we champion the use of oracle-based evaluation. The central theme of this thesis is an appeal to look at all aspects of PP attachment in more realistic terms, and more realistic evaluation is certainly conducive to that goal. We do observe a degradation in performance moving from oracle- to parser-based evaluation, but it is important to note that this is a difference of degree, not of kind. Oracle-based evaluation can allow inferences about the real performance of attachers so long as we keep in mind that some drop in performance should be expected when moving to more realistic scenarios, and, whenever possible, gauge the expected degree of that drop.

3.2.2 Feature Realism

Being aware of any unrealistic preprocessing advantages, if not avoiding them altogether, is equally important when looking beyond the canonical view of attachment and incorporating additional contextual information. There are inevitably many types of information that may be invaluable to any given NLP task that have yet to be codified into any knowledge base of sufficient completeness and accessibility, and an infinitude of automated tools to generate or extract useful information that have yet to be developed. This should not preclude a theoretical or proof-of-concept assessment of the usefulness of new sources of information. Indispensable tools and resources might never be built without first establishing a need for them. Nonetheless, when experimenting with and evaluating such new features using manually annotated information, realistic evaluation depends on an awareness of how realistic or unrealistic these features are in terms of the plausibility of a real-world automated information source, the level of accuracy that might be expected in reality, and whether degradation of accuracy moving from a “proof-of-concept oracle” to a realistic implementation affects the usefulness of the new feature in degree or in kind.

One area where a “proof-of-concept oracle” has shown promise for significantly improving PP attachment is the use of semantic role information. As we have discussed in Section 1.2, semantic roles and PP attachments are very tightly intertwined in that the latter often specifies the participants in a given relation while the former specifies the type of that relation. In theory, knowing the type of the relation helps in identifying the participants, and vice versa. This is borne out in the literature, where features encoding manually annotated semantic role information have been shown to accurately predict attachment. But how realistic are these features?

Mitchell (2004) experiments with exploiting semantic role information using the function tags included in the manual annotations of the Penn Treebank. These function tags encode super-syntactic information about phrases, including in the case of some phrases, their semantic role. Thus the phrases in the following sentences, for example, are annotated with function tags indicating that they are involved in a `DIRECTION` relation in the case of Example (3.4a), and a `BENEFACTIVE` relation in the case of Example (3.4b).

(3.4) a. *I flew* [$_{\textit{PP-DIRECTION}}$ *from Tokyo*] [$_{\textit{PP-DIRECTION}}$ *to New York*].
b. *I baked a cake* [$_{\textit{PP-BENEFACTIVE}}$ *for Doug*].

Mitchell compares the performance of several different machine learning algorithms with and without the additional function tag information. He observes an enormous margin of improvement with the addition of function tags, from roughly 85% accuracy without function tags to upwards of 92% with them, depending on the learning algorithm used.

Unfortunately, it is not known whether such impressive results are attainable without a function-tag oracle. There have been efforts toward automated tagging of PTB-style function tags (Blaheta and Charniak, 2000; Merlo and Musillo, 2005), and more generally, semantic role labeling is a popular area of inquiry (Carreras and Màrquez, 2005). However, to our knowledge, it has yet to be shown whether any such tools are capable of improving PP attachment to a similar degree.

Olteanu and Moldovan (2005) also experiment with using semantic role information from a manually annotated source: FrameNet[5] (Ruppenhofer et al., 2006), a manually compiled lexical database containing semantic frames. Semantic frames describe an event,

[5] `http://framenet.icsi.berkeley.edu/`

relation, or entity along with the relevant participants. For example, the semantic frame for Example (3.5) is `Contacting`, which is evoked by the verb *wrote*.

(3.5) John wrote to Mary about syntax.

The participants are *John*, with the semantic role of `Communicator`, and *Mary*, with the semantic role of `Addressee`.

In their experiments, Olteanu and Moldovan compare the performance of a support vector machine with and without semantic features derived from semantic frames. Specifically, they use the name of the semantic frame evoked by the verb attachment candidate and the semantic role of the noun attachment candidate. They observe a noticeable improvement from their semantic features, as tested on their own corpus (Olteanu, 2004) extracted from FrameNet's annotated example sentences. However, it is unclear to what degree these results correspond to reality as the corpus is rather artificial in nature, and may suggest drastically different coverage and frequency of phenomena than what would be observed in reality. Rather than representing all naturally occurring attachment ambiguities within a collection of texts, their corpus is limited to occurrences that provide evidential support for particular frame annotations. Crucially, this entails that by the very construction of the corpus, the relevant semantic information from FrameNet for every attachment instance is guaranteed to be available. FrameNet is by no means a complete reference on the semantic roles of all terms that can occur in real text, nor is it meant to be. Accordingly, the semantic features, as investigated by Olteanu and Moldovan, would be available in only a fraction of attachment candidates in naturally occurring PP ambiguities.

Note that the unrealistic advantage afforded by the semantic-role-based features of both Mitchell and of Olteanu and Moldovan is qualitatively different from that of the quadruple extraction oracle admonished by Atterer and Shütze. As we have seen in the previous chapters, resolution of attachment ambiguity can be equivalent to determining relevant semantic roles in some cases. If we can be certain, say, that [$_{PP}$ *with a telescope*] specifies an `INSTRUMENT` relation, the attachment ambiguity in the classical example *"I saw a man with a telescope"* effectively disappears. Thus, here it is not merely a question of using relevant information that would be unavailable or much less accurate or relevant in a more realistic text processing scenario. A semantic role oracle is not merely helpful to an unrealistic degree, it is actually giving away some of the answers. In Chapter 5 we report on our own experiments with semantic-role-based features in an attempt to get a more accurate sense of their practicality and potential in realistic processing tasks.

3.3 Evaluation Guidelines

We have discussed the importance of appropriate baselines and avoiding, or at least assessing the impact of, unrealistic input. In this section we outline the corresponding evaluation choices we make and the guidelines we will follow in evaluating our experiments throughout this thesis.

A key consideration that emerged from the above discussions of both appropriate baselines and input realism is the relevance of selecting a suitable data source. In particular, it should seem obvious now that the RRR corpus, despite having been an invaluable resource in its time, is no longer adequate given that:

- it does not hold up to the high standards of annotation quality and correctness of modern alternatives;

- because it is not a faithful representation of its underlying text, it cannot be readily applied to any task that does not fit the narrow definition for which the corpus was designed; and
- it cannot support any inquiry beyond the canonical form of ambiguity and context (i.e. lexical quadruples).

In our experiments throughout this thesis, we use the WSJ corpus from the latest version of the Penn Treebank (PTB-3).[6] Whereas the preliminary version 0.5—from which the RRR corpus is extracted—represents an exploratory probe into the possibility of building a large-scale treebank, PTB-3 reflects the state of the art after several iterations of development. Annotation guidelines are well documented and consistently applied, and the errors and inconsistencies present in earlier versions have been corrected. As such, the annotation errors causing almost 20% of the quadruple extraction failures observed in our previous experiment on the RRR corpus are all but eliminated.

Using the newer PTB-3 addresses our first issue with the RRR corpus. Simply rebuilding the RRR corpus with the improved source data would allow more realistic attachment evaluation on its own. However, we would also like to look beyond the canonical view of attachment, and thus the latter two issues with the RRR corpus are more of concern. With direct access to the actual text of our corpus, we are unconstrained in looking at any non-canonical ambiguities or any combination of additional context beyond lexical tuples. But, just as the choices and assumptions of Ratnaparkhi et al. in their framing of the problem do not accommodate our information and evaluation needs, our view of attachment will almost certainly differ from that of some future inquiry. A natural language sentence, or a parse thereof, is a remarkably complex and dense representation of information, only some of which is relevant to our concerns in PP attachment. Inevitably, we will need to use intermediate representations of the sentences we process, such as the canonical attachment quadruples or other feature representations. Any intermediate representation will necessarily reflect the details and assumptions of our particular framing of the problem. Nonetheless, if any representations extracted from the treebank text (such as attachment quadruples) maintain a faithful representation of the underlying text, we can ensure that our results remain comparable to future inquiries.

By using the WSJ text directly instead of RRR quadruples, we lose one of the main advantages of the RRR corpus: the ability to compare performance with a wide assortment of attachment techniques cheaply. However, this ease of comparability is actually a false economy, as these classical attachment techniques really cannot serve as appropriate baselines. As we have discussed, a meaningful evaluation requires comparison against a parser baseline, but this is more cumbersome, and not entirely accurate, on the RRR corpus, as we have seen in our re-hashing of Atterer and Schütze's experiments in the previous section. The WSJ corpus, on the other hand, is the de facto standard treebank for training and testing constituency parsers. Thus, what is lost in ease of comparison with classical attachment techniques is gained in ease of comparison with parser baselines.

We have chosen to use Charniak's parser (Charniak, 2000) both as our primary evaluation baseline and as our source of input for the various attachment techniques that we look at throughout this thesis. This is not a principled choice; a wide variety of parsers are available employing all manner of strategies, and most of these would serve as suitable

[6] In Chapter 6, we use the GENIA Treebank (GTB), which offers a similar level of high quality syntactic annotation for texts in the biomedical domain.

baselines. Charniak's parser is, however, quite widely used and among the most accurate. This is not to say that an attachment technique must beat the parser with the best attachment performance in order to be useful. There are many different parsers that excel in different aspects of parsing. An attacher that can improve upon a parser with abysmal attachment performance, but that is particularly strong in other aspects, may result in a superior overall parsing package. For example, in our preliminary experiments we observed that several attachment techniques that did not offer a statistically significant improvement over Charniak's parser did yield an improvement over the Stanford parser, with both unlexicalized model (Klein and Manning, 2003a) and factored model (Klein and Manning, 2003b), and the Berkley parser (Petrov and Klein, 2007).

In addition to the attachment performance of Charniak's parser, which we use as a baseline in our evaluations, we will also contrast results with the attachment performance of Charniak's reranking parser (Charniak and Johnson, 2005). This is simply Charniak's base parser with an additional postprocessing stage that discriminatively reranks the top n best parses. Discriminative parse reranking (Collins, 2000) is a technique that reanalyzes the output of a probabilistic parser in an attempt to select more accurate parses. The reranker provides a complementary view of the data, allowing parse trees to be considered in terms of arbitrarily complex features that need not be constrained to local context, and which would be difficult to incorporate into a generative model. In a sense, the reranker can be seen as performing a similar function to PP attachers on a more general level: discriminatively selecting among multiple possible parses of an ambiguous sentence. As such, we choose to view the reranking parser not as a minimum baseline but rather as a competing alternative approach to improving PP attachments.

Chapter 4

Beyond Binary Ambiguity

In our description of the canonical form of PP attachment in Section 1.3, we noted that work on prepositional phrase attachment has almost always been limited to ambiguity cases involving a binary decision between verb and noun attachment. Perusal of the literature may lead to the assumption that other types of attachment have no—or trivial—ambiguity, occur infrequently, or are otherwise uninteresting. This is a simplification that does not adequately represent the task of PP attachment outside of artificially constructed data sets. Instead, real-world texts contain many more types of PP attachment ambiguity other than the traditional V/N distinction. In the WSJ corpus, for example, 56.2% of PPs that have V/N ambiguity also have multiple noun attachment possibilities (hereafter, V/N$^+$ ambiguity). In addition to multiple nouns, attachment possibilities may also include multiple verbs in the case of reduced relative clauses, as in Example (4.1a) below, adjectival or adverbial phrases, as in Example (4.1b) or any combination thereof.

(4.1) *a. I [$_{verb}$ saw] the man [$_{verb}$ feeding] birds in the park [$_{PP}$ with the telescope].*
b. Binoculars are [$_{adj}$ similar] in function [$_{PP}$ to telescopes].

Ignoring these more elaborate possibilities for the moment and looking only at the least complex ambiguity case after V/N, those with a verb attachment candidate and multiple noun attachment candidates (V/N$^+$), the increase in complexity is substantial. The additional noun attachment candidates come from any PPs that occur between the verb and the PP under consideration, as in:

(4.2) *Environmentalists are [$_{verb}$ pushing] for [$_{noun}$ barriers] to any future [$_{noun}$ imports] of [$_{noun}$ oil] from the Canadian tar [$_{noun}$ sands] [$_{PP}$ into the European market].*

Whereas a single ambiguous PP has two possible syntactic interpretations, the number of interpretations for a "chain" of several consecutive PPs grows combinatorially[1] such that a chain of two PPs has five interpretations, and a chain of three has fourteen. Even if we simplify matters by considering each PP attachment independently, the possibilities are substantial. Since any number of PPs can occur consecutively, there is no hard limit to the number of attachment candidates for each PP.

Whereas canonical V/N attachment deals with 4-tuples, V/N$^+$ attachment is concerned with $k+3$-tuples of the form:

$$(v, n_1, \ldots, n_i, \ldots, n_k, p, n_c),$$

[1] The number of possible syntactic interpretations for a chain of n consecutive PPs is the $(n+1)^{\text{th}}$ Catalan number (Church and Patil, 1982), where the i^{th} Catalan number is defined as $C_i = \frac{1}{i+1}\binom{2i}{i}$.

where k is the number of noun attachment candidates. The correct attachment, A, of each such tuple is specified as $A \in \{V, N_1, \ldots, N_i, \ldots, N_k\}$, where V denotes attachment to the verb and N_i denotes attachment to the i^{th} noun candidate.

Recalling the backed-off model of Collins and Brooks (1995), sparsity concerns increase with tuple size. Thus classification of $k+3$-tuples, where k can be upwards of eight or nine, will be much more susceptible to sparsity. Collins and Brooks observed that 8.8% of the quadruples in their test set were also present in their training set, or equivalently, that 8.8% of their test set could be attached without any backing-off. In our data, we observe a similar 8.2% of test quadruples occurring in the training set. When looking at 5-tuples—cases with just one additional noun attachment candidate—we observe only 2.5% of test tuples in the training set. Further, none of the higher dimensional tuples in our test set occur in the training set at all. Unfortunately, backing off to lower-order tuples is neither as straightforward nor as effective with k+3-tuples as it is with 4-tuples, as will be discussed in the following section.

Even where these concerns can be minimized, the techniques and/or features that perform best at resolving binary ambiguity may not be optimal or appropriate for higher dimensional ambiguity. One experiment (Franz, 1996) demonstrates this quite clearly in comparing a multi-featured[2] log-linear approach with the classic lexical association model of Hindle and Rooth (1993) on both a V/N task and a task with one additional noun candidate (V/N^2). On the V/N task, the multi-featured model provides no significant improvement over the lexical association model. However, it gives a marked improvement (79% versus 72% accuracy) when evaluated over PPs with V/N^2 ambiguity. Further, the optimal feature set differs between the two experiments.

In this chapter, we present an extension to the backed-off model (Collins and Brooks, 1995) to non-binary V/N$^+$ ambiguous PP attachments. Our goal here is to explore some of the key difficulties that arise in scaling up to higher-ambiguity attachment cases and to develop a better understanding of the problem space. This informs on the choice of the backed-off model as our starting point and of limiting our extension to V/N$^+$ ambiguities (rather than V$^+$/N$^+$, or indeed the full spectrum of possible attachment ambiguities). We have seen in Chapter 3 that Collins and Brooks' backed-off model does not outperform modern parsers, which consider similar contextual features with more sophisticated machinery; it is unlikely to fare much better on a more difficult class of ambiguity cases. However, the simplicity of the backed-off model allows an exceptional level of clarity and understanding of the differences between resolving binary and higher-ambiguity attachment. It is thus much better suited to our current purpose than more sophisticated alternatives.

4.1 Extending the Backed-Off Model

There are a number of ways to deal with the additional noun attachment candidates in V/N$^+$ ambiguous cases. Perhaps the simplest solution is to do away with as much of the additional ambiguity with as little consideration as possible. Applying the principle of right association, we could eliminate the additional noun candidates by always choosing the lowest one. The problem is then reduced to a V/N decision between the verb and lowest noun candidate, and we can use Collins and Brooks' model as is. This is certainly a good

[2] Features included the preposition, lexical association scores for each attachment candidate, part-of-speech tags of noun attachment candidate and prepositional complement, and an indication of the definiteness of the noun candidate.

starting baseline, but we should be able to do better by considering lexical data from the noun candidates. Recall from Chapter 2 that lexical association is a much better predictor of attachment behavior than purely structural principles.

With potentially so many attachment possibilities, it may be tempting to simplify the task by relaxing constraints and assuming independence between candidates. The likelihood of attachment to each candidate could be assessed irrespective of any other attachment candidates, and the candidate with the highest likelihood could be selected for attachment in a one-versus-all fashion. An MLE model adopting such a strategy might look like:

$$\underset{x \in \{v, n_1, \ldots, n_i, \ldots, n_k\}}{\text{argmax}} \frac{f(x, p, n_c)}{f(x)}.$$

However, this assumption of candidate independence is essentially premature backing off. Given that Collins and Brooks demonstrate quite clearly that even low-occurrence higher-order tuples are better than lower-order tuples, discarding available higher-order tuples without even looking at them is a losing proposition.

In one of very few studies to look at non-binary PP attachment ambiguities, Merlo, Crocker, and Berthouzoz (1997) give an extension of Collins and Brooks' model to cases with one, two, or three noun candidates. Their model applies maximum likelihood estimates directly to high-order tuples, backing-off as necessary. Crucially, to address the increased sparsity of higher-order tuples, they incorporate statistics from binary attachment data.

We present a similar approach here to extend the backed-off model to ambiguity cases with any number of noun candidates. Like Merlo et al., we address the concerns of increased sparsity by framing the decision problem in terms of binary sub-problems. However, given the degree of sparsity of higher-order tuples, we see maximum likelihood estimates directly on these tuples as less than ideal: the data simply does not occur in our training data. Instead, the approach that we propose here decides attachments with multiple noun candidates (V/N$^+$) by mapping k+3-tuples directly into 4-tuples and performing multiple binary decisions using Collins and Brooks' backed-off estimation procedure, as described in Chapter 2. A slight reformulation is given in Algorithm 4.1 to reflect that each binary decision may be between two nouns as well as between a noun and a verb. Specifically, the input quadruple is represented as (h, l, p, n_c), where h and l are the high (further from the preposition) and low (closer to the preposition) attachment candidates, respectively. Similarly, wherever the correct attachment is known, it is specified as H or L instead of V or N.

Information from each binary decision is combined in a one-versus-one (max wins) fashion, as outlined in Algorithm 4.2. First, each noun candidate is compared to every other noun candidate, using the binary procedure as described in Algorithm 4.1. The candidate that "wins" the largest number of these binary comparisons is selected as the best noun candidate. This noun is in turn compared with the verb candidate to decide the final attachment.

Our extended model needs to make binary noun-noun (N/N) comparisons in addition to V/N comparisons (another distinction from the model of Merlo et al., where no direct noun-noun candidate comparisons are made, biasing heavily toward V/N relationships). While a 4-tuple representing an N/N attachment ambiguity, (n_H, n_L, p, n_c), may be superficially quite similar to a V/N tuple, subtle differences (which we discuss in the following section) require that we build separate models for the V/N case and the N/N case.

Both models consider 4-tuples representing binary ambiguities, but in training them we would like to use all available data, including k+3-tuples like those we wish to disambiguate.

Algorithm 4.1 Generalized binary backed-off estimation procedure

```
procedure ESTIMATE-LOW-ATTACHMENT-PROBABILITY($h, l, p, n_c$)
    if $f(h, l, p, n_c) > 0$ then
        $P(L|h, l, p, n_c) \leftarrow \frac{f(L, h, l, p, n_c)}{f(h, l, p, n_c)}$
    else if $f(h, l, p) + f(h, p, n_c) + f(l, p, n_c) > 0$ then
        $P(L|h, l, p, n_c) \leftarrow \frac{f(L, h, p, n_c) + f(L, l, p, n_c) + f(L, h, l, p)}{f(h, p, n_c) + f(l, p, n_c) + f(h, l, p)}$
    else if $f(p, n_c) + f(h, p) + f(l, p) > 0$ then
        $P(L|h, l, p, n_c) \leftarrow \frac{f(L, p, n_c) + f(L, h, p) + f(L, l, p)}{f(p, n_c) + f(h, p) + f(l, p)}$
    else if $f(p) > 0$ then
        $P(L|h, l, p, n_c) \leftarrow \frac{f(L, p)}{f(p)}$
    else
        $P(L|h, l, p, n_c) \leftarrow 1$
    end if
    return $P(L|h, l, p, n_c)$
end procedure
```

Algorithm 4.2 Extended backed-off attachment procedure

```
procedure DECIDE-ATTACHMENT($v, n_1, \ldots, n_i, \ldots, n_k, p, n_c$)
    // Compare all noun candidates.
    for $i = 1 \rightarrow (k - 1)$ do
        for $j = i + 1 \rightarrow k$ do
            if ESTIMATE-LOW-ATTACHMENT-PROBABILITY($n_i, n_j, p, n_c$) $\geq 0.5$ then
                $wins_j$++
            else
                $wins_i$++
            end if
        end for
    end for

    // Select best noun candidate.
    $best \leftarrow \underset{i \in 1,\ldots,k}{\text{argmax}}(wins_i)$
    // Compare verb candidate with best noun candidate.
    if ESTIMATE-LOW-ATTACHMENT-PROBABILITY($v, n_{best}, p, n_c$) $\geq 0.5$ then
        return $n_{best}$
    else
        return $v$
    end if
end procedure
```

We thus require a way to map $k+3$-tuples into corresponding 4-tuples. Describing this mapping would benefit from an illustrative example, for which we refer to Example (4.2), repeated here with its corresponding 7-tuple:

> *Environmentalists are* $[_{verb}$ *pushing*$]$ *for* $[_{noun}$ *barriers*$]$ *to any future* $[_{noun}$ *imports*$]$ *of* $[_{noun}$ *oil*$]$ *from the Canadian tar* $[_{noun}$ *sands*$]$ $[_{PP}$ *into the European market*$]$.
>
> ⇓
>
> *(4.3)* $(N_2, pushing, barriers, imports, oil, sands, into, market)$

In order to make use of the binary backed-off estimation procedure in Algorithm 4.1, the semantics of each 4-tuple that we extract from a $k+3$-tuple must be consistent with the semantics expected by the backed-off procedure. Mapping between $k+3$- and 4-tuples is therefore not simply a matter of extracting all possible 4-permutations; only some of these are valid, namely those that adhere to the following three basic constraints:

1. **Each 4-tuple must represent one PP and two of its attachment candidates.** This constraint should hopefully seem obvious as simple definitional aspect of the backed-off procedure. It ensures that all 4-tuples represent a binary PP attachment ambiguity. For Example (4.3), this constraint dictates that 4-tuples of the form $(x, y, into, market)$ are valid, as these represent binary PP attachment ambiguities, but 4-tuples such as $(pushing, barriers, imports, oil)$ are not valid, since *imports* is not a preposition and *oil* is not a prepositional complement and the tuple thus does not represent a PP attachment ambiguity.

2. **The elements of each 4-tuple must maintain the same relative order as in the original $k+3$-tuple.** This constraint ensures that the attachment candidates, preposition, and its complement are identifiable based on their order in the tuple, and that the order of the two candidates in the 4-tuple corresponds to their order in the underlying text. For Example (4.3), this constraint dictates that $(imports, oil, into, market)$ is valid, but $(oil, imports, into, market)$ is not, since it represents a word ordering that does not correspond to that of the underlying sentence.

3. **Each 4-tuple must include the correct attachment site as one of the candidates.** This constraint ensures that each 4-tuple states a relative preference for one attachment over the other. For Example (4.3), we can extract 4-tuples that give evidence concerning the relative attachment likelihood of the actual attachment site, *imports*, versus any of the other attachment candidates, such as $(imports, oil, into, market)$. But what about the other pairwise comparisons that can be made between attachment candidates? The original 7-tuple specifies that neither *oil* nor *sands* are the actual attachment site in this sentence, but it provides no indication of the relative likelihood (or unlikelihood) of attachment to either candidate, therefore $(oil, sands, into, market)$ is not a valid 4-tuple in this case. One of these options may be a more or less likely attachment site than the other, but we are unable to tell from the information in this particular 7-tuple.

In all, the 4-tuples that can be extracted from Example (4.3) according to the given constraints are as follows:

V/N model 4-tuples	N/N model 4-tuples
$(L, pushing, imports, into, market)$	$(L, barriers, imports, into, market)$
	$(H, imports, oil, into, market)$
	$(H, imports, sands, into, market)$

Lower-order tuples for back-off stages are counted in essentially the same way as in the original backed-off model. All possible sub-tuples of all extracted 4-tuples are included in both V/N and N/N models, so long as they include the preposition. However, a sub-tuple is only counted once per source $k+3$-tuple. For example, two of the N/N 4-tuples that we have extracted from our example $k+3$-tuple yield the same backed-off triples, doubles, and singles:

$$\begin{array}{c} (H, imports, oil, into, market) \\ (H, imports, sands, into, market) \end{array} \Rightarrow \begin{array}{c} (H, imports, into, market) \\ (H, into, market) \\ (H, imports, into) \\ (H, into) \end{array}$$

These backed-off tuples are counted only once, not separately for each 4-tuple.

A full outline of the mapping of $k+3$-tuples into lower-order tuples for both the V/N and N/N models is provided in Algorithms 4.3 and 4.4, respectively.

Algorithm 4.3 Generating V/N sub-tuples from $k+3$-tuple training instances

```
procedure GENERATE-VN-SUBTUPLES($A, v, n_1, \ldots, n_i, \ldots, n_k, p, n_c$)
    if $A = V$ then
        for all $n_i$ do
            Add the following tuples to the set of V/N training tuples:
            $(H, v, n_i, p, n_c), (H, n_i, p, n_c), (H, v, n_i, p), (H, n_i, p)$
        end for
        Add the following tuples to the set of V/N training tuples:
        $(H, v, p, n_c), (H, v, p), (H, p, n_c), (H, p)$
    else if $A = N_a$ then
        Add the following tuples to the set of V/N training tuples:
        $(L, v, n_a, p, n_c)$,
        $(L, v, p, n_c), (L, n_a, p, n_c), (L, v, n_a, p)$,
        $(L, n_a, p), (L, v, p), (L, p, n_c)$,
        $(L, p)$
    end if
end procedure
```

4.2 Representational Concerns

While the approach we have described for deciding V/N$^+$ attachment ambiguities in terms of binary comparisons is quite simple, building the necessary models—i.e. mapping $k+3$-tuples into 4-tuples—requires more consideration. A major reason for this is that the representational elegance of the 4-tuple is largely lost in higher dimensional tuples. When

Algorithm 4.4 Generating N/N sub-tuples from k+3-tuple training instances

```
procedure GENERATE-NN-SUBTUPLES($A, v, n_1, \ldots, n_i, \ldots, n_k, p, n_c$)
    if $A = N_a$ then
        if $a > 1$ then        // There are higher alternatives to the correct attachment point.
            for $i = 1 \rightarrow (a-1)$ do
                Add the following tuples to the set of N/N training tuples:
                $(L, n_H = n_i, n_L = n_a, p, n_c), (L, n_H = n_i, n_L = n_a, p),$
                $(L, n_H = n_i, p, c), (L, n_H = n_i, p)$
            end for
            Add the following tuples to the set of N/N training tuples:
            $(L, n_L = n_a, p, c), (L, n_L = n_a, p), (L, p, c), (L, p)$
        end if
        if $a < k$ then        // There are lower alternatives to the correct attachment point.
            for $i = (a+1) \rightarrow k$ do
                Add the following tuples to the set of N/N training tuples:
                $(H, n_H = n_a, n_L = n_i, p, n_c), (H, n_H = n_a, n_L = n_i, p),$
                $(H, n_L = n_i, p, c), (H, n_L = n_i, p)$
            end for
            Add the following tuples to the set of N/N training tuples:
            $(H, n_H = n_a, p, c), (H, n_H = n_a, p), (H, p, c), (H, p)$
        end if
    end if
end procedure
```

looking at the canonical attachment problem, we have presented the distinction between the two attachment options primarily as an issue of lexical preference; but it can also be a structural distinction (high versus low, minimal attachment versus right association), a semantic distinction between conceptual classes of words, a categorical distinction between verb and noun, and often an ontological distinction between event and entity. All of these aspects are neatly rolled in to any decisions we make about V/N attachments, but these must be teased apart to some degree in order to permit a meaningful mapping to 4-tuples from higher-order tuples and to maintain the semantics of backing off with N/N tuples.

The first issue is that the aspect of lexical category preference, which is present when comparing verb and noun attachment candidates, is obviously not present when comparing two attachment sites that are both nouns. Knowing that a given preposition tends toward either verb attachment or noun attachment does not help in determining its lexical preference between two given nouns. Conversely, knowledge of a given preposition's preference for attachment to a particular noun over another given noun provides no basis for determining its preference for noun or verb attachment in general.

There is also the issue of the decoupling of structural and lexical preference in N/N tuples. Consider the 7-tuple from Example (4.3):

$$(N_2, \textit{pushing}, \textit{barriers}, \textit{imports}, \textit{oil}, \textit{sands}, \textit{into}, \textit{market}).$$

In terms of lexical preference among noun candidates, we can compare the actual attachment site *imports* with all other alternatives. However some of these are structurally higher than *imports* and some are structurally lower. We have evidence for [$_{\text{PP}}$ into market] preferring both low and high attachment:

$$(L, barriers, imports, into, market)$$

$$(H, imports, oil, into, market)$$

Does the fact that the correct attachment is higher than two alternatives and lower than only one suggest that a high structural preference applies in general? Based on this particular training instance, what decision would be best if faced with disambiguating the tuple $(imports, barriers, into, market)$, where our prior knowledge of the lexical preference of *imports* over *barriers* suggests a different decision from our prior knowledge of structural preference?

Both these issues demonstrate the need for maintaining separate V/N and N/N models. More importantly, they illustrate how much more difficult a problem it is to resolve V/N$^+$ ambiguities compared to V/N ambiguities. Here, the various aspects of attachment do not reinforce each other allowing greater generalization as they do in the canonical backed-off model. Instead, the probability mass of training instances must be split along several dimensions.

Absolute structural preference is another aspect in which V/N comparisons differ from N/N comparisons. With the former, the verb candidate v is always the highest, or most distant, attachment candidate, so each V/N 4-tuple maintains some notion of the absolute position of its candidates in the source k+3-tuple. In N/N 4-tuples, only the relative structure between the two candidates is maintained in the mapping; given a 4-tuple (n_H, n_L, p, n_c), we can tell that n_H occurs above n_L in the source $k+3$-tuple, but not whether either or both candidates are the highest or lowest alternatives, or somewhere in the middle. Consider again the k+3-tuple from Example (4.3). Without any notion of absolute structure, we can make no structural distinction between the following sub-tuples:

$$(barriers, sands, into, market)$$

$$(imports, oil, into, market).$$

The first tuple represents a choice between two structural extremes—*barriers* is the highest/farthest noun candidate and *sands* is the lowest/closest. In contrast, the two noun candidates in the latter tuple are comparatively close together, and neither represents a structural extreme. That this difference is not expressible in our extended model may reflect a significant loss in representational adequacy.

Gibson et al. give evidence that an important distinction could be made here. They analyze ambiguous attachments involving three noun attachment candidates, showing through corpus analysis (Gibson and Pearlmutter, 1994) that attachment to the middle candidate is a much rarer occurrence than attachment to either the highest or lowest candidates. In reading experiments (Gibson et al., 1996), they observe that human subjects take longer to process attachments to the middle candidate than either high or low attachments, and that they are much more likely to consider middle attachments as ungrammatical. Gibson et al. offer as explanation of this phenomenon the interplay between the competing principles of recency and predicate proximity. Recency specifies an attachment preference for more recent words in the input stream, similar to right association. Looking again at our example $k+3$-tuple, the most likely attachment site according to the recency principle is *sands*, followed in order by *oil*, *imports*, *barriers*, and *pushing*. Predicate proximity specifies a preference for attachments to be as close as possible to the head of a predicate. The predicate in Example (4.2) is *are pushing*, and thus according to the principle of predicate proximity, the most likely attachment site is the head of the predicate itself, *pushing*. The remaining attachment candidates decrease in attachment likelihood in order from *barriers* to *sands*. For the highest and lowest attachment candidates, recency and predicate proximity have contradictory preferences. Assuming both principles carry equal weight, the net effect is that both *pushing* or *sands* are somewhat likely attachment sites. However, neither recency nor predicate proximity express a preference for the intermediate candidate *imports*, and it is thus a much less likely attachment site. In essence, predicate proximity pulls attachment preference toward higher alternatives while recency pulls attachment preference toward lower alternatives, leaving any intermediate attachment alternatives relatively unsuitable candidates.

4.3 Results

We use the Penn Treebank Wall Street Journal corpus for training (sections 2-21) and testing (section 23). Head-word tuples along with their correct attachments are extracted for each ambiguous PP having a potential verb attachment site and at least one potential noun attachment site. The extraction process (detecting PPs, their attachment candidates, and their actual attachment) relies directly on the gold-standard parses from the treebank. The resulting data is meant to be similar to the RRR corpus (Ratnaparkhi, Reynar, and Roukos, 1994), but with k+3-tuples instead of 4-tuples. A total of 43 384 tuples are extracted for training, and 2259 for testing. Among these, 9599 training tuples and 501 test tuples have $k > 1$. For comparison, the RRR corpus contains 20 804 training and 3097 testing quadruples.

To highlight the impact of the increase in ambiguity, we evaluate our extended back-off model on both the full V/N$^+$, as well as the binary V/N sub-task. Note that on the binary task, our extension to the backed-off model is not applicable and the model is functionally equivalent to the original backed-off model. This allows us a point of comparison between our tuple data and the RRR corpus. In Chapter 3, we reported an attachment accuracy of 84.3% for our implementation of the backed-off model on the RRR corpus. On our data extracted from the WSJ corpus, we observe an accuracy of 86.56%. The difference can be attributed to roughly 60% more training data. Our V/N data includes 33 785 training quadruples versus 20 804 training quadruples in the RRR corpus. Some of the difference may also be due to the higher quality of our training data, as it is not subject to the extraction errors discussed in Chapter 3.

The attachment accuracy for our extended back-off model is given in Table 4.1, along with several baselines for comparison. Again, we use Charniak's parser (both base parser and reranking parser accuracies are shown) as a baseline. Overall, our proposed approach performs worse than the parser, though this is as expected from the relative performance between the original backed-off approach and parser on the binary task.

A significant decrease in attachment accuracy between the binary V/N case and the V/N$^+$ case is observed for both our extended model and the parser. This is also not surprising given the significant increase in complexity between the two tasks, and the higher degree of sparsity from dealing with higher-context data (bigger tuples) and from the decoupling of discriminating factors (e.g. lexical preference, structural preference, category).

Both surprising and rather disappointing is the performance of the extended model when compared to the naive baseline of ignoring additional noun candidates and merely

Table 4.1: Attachment accuracy (%) of extended backed-off model

	V/N$^+$	V/N
Extended back-off	83.73	86.56
Naive baseline (v vs n_k)	83.24	-
Base parser	84.83	86.50
Reranking parser	87.24	89.01
# of instances	2259	1758

applying the binary backed-off model to the verb and lowest noun candidate. While the extended model shows a slight improvement over the baseline, the difference is not statistically significant. Despite ample evidence for the greater predictive power of lexical information over structural principles, our N/N model, using all available lexical preference information, is unable to significantly outperform simple and blind application of a purely structural principle.

One possible issue that may contribute to this lackluster result is that of absolute structure and the preference for attachment to the high or low extremities over middle candidates, as discussed in the previous section. It would be interesting to explore this dimension further within the framework of the (extended) backed-off model, perhaps by differentially weighting candidates based on distance from the bottom and top of the structure, but we leave this to future work.

Chapter 5

Beyond Lexical Association

Our outline of the canonical form in Section 1.3 presents PP attachment as a head-to-head relationship. In Chapter 2, we introduced several successful attachment approaches that adhere to this view, leveraging lexical associations between the relevant head words v, n, p, and n_c. Yet, we know that this view is a simplification. Other features could be helpful, or even essential in deciding some attachments.

We have seen that prenominal modifiers and determiners of the prepositional complement can bias our attachment decision, as in the following example copied from Chapter 1:

(5.1) *a. I saw the man with my own eyes.*
b. I saw the man with blue eyes.

In the following example, attachment of both PPs is informed by the prenominal modifiers of their attachment sites rather than the heads of those phrases.

(5.2) *a. Many commercial light trucks carry* $[_{adj}$ *more*$]$ *people* $[_{PP}$ *than cargo*$]$ *and therefore should have the* $[_{adj}$ *same*$]$ *safety features* $[_{PP}$ *as cars*$]$.
b. Saatchi is struggling through the $[_{adv}$ *most*$]$ *troubled period* $[_{PP}$ *in its 19-year history*$]$.

Hindle and Rooth (1993) note that NP attachment candidates with superlative prenominal modifiers, like *most*, invariably indicate noun attachment in their data.

Human readers exploit a wealth of information beyond the four head words of the canonical model when making attachment decisions. An experiment measuring the attachment performance of three human treebanking experts on a random selection of the RRR corpus demonstrates this quite clearly (Ratnaparkhi, Reynar, and Roukos, 1994). When given only the four head words, the human experts averaged 88.2% attachment accuracy. Given the full text of the underlying sentences, their performance was markedly improved to 93.2% accuracy. Hindle and Rooth (1993) similarly report on the discernibly increased difficulty of making attachment decisions based solely on head words. Prior to annotating their test data, they recorded their own attachment decisions using only the same head-word context available to their system, averaging only 86.4% accuracy between them.

Undoubtedly, there is useful context that is ignored by attachment methods looking only at head-word quadruples. In this chapter, we look at exploiting a larger and more varied set of features.

5.1 Olteanu & Moldovan's Large-Context Model

Applying context other than lexical association of head words toward automated resolution of PP ambiguity is not a revolutionary new approach. In Section 2.2 we recounted several approaches that apply some notion of semantic similarity toward attachment resolution, using lexical resources like WordNet or statistical semantics. Structural principles, like right association, are also incorporated into most attachment approaches. However, few approaches attempt to integrate all linguistic knowledge that can be useful for attachment, likely in part because much of it cannot be gleaned from the prevalent RRR quadruples.

Olteanu and Moldovan (2005) provide a noteworthy exception in this respect. Their "large-context model" combines a diverse set of lexical, syntactic, and semantic features using a support vector machine. Our experiments in this chapter are based on their features. The best performing among them are outlined below.

5.1.1 Head-based Features

Head-word quadruples offer a simple and effective (though incomplete) model of PP attachment, and are an obvious starting point for building a more comprehensive model. The large-context feature set includes the following features, which encode the quadruple elements as well as some familiar variations for the sake of smoothing:

v, n, p, n_c: The standard four head words in their surface form—i.e. exactly as they occur in the text.

$\{v,\ n,\ n_c\}$`-pos`: The part of speech of the verb and noun candidates and the prepositional complement.

Bear in mind here, that although n`-pos` and n_c`-pos` are the heads of NPs, they are not necessarily nouns. Among others, NPs can be headed by pronouns, as in

$[_{NP}\ [_{pronoun}\ \textit{they}]]$ *sent* $[_{NP}\ [_{pronoun}\ \textit{him}]]$ *on a wild goose chase.*

Pronouns generally cannot accept attachments, so the ability to distinguish them from nouns is quite useful.

Also, the part-of-speech tags used here are finer-grained than just verb and noun. There are several different verb tags, for instance, including base form verb, past tense verb, gerund or present participle, and past participle. Noun sub-tags distinguish between singular and plural nouns and common and proper nouns.

$\{v,\ n,\ n_c\}$`-lemma`: The lemma of the verb and noun candidates and of the prepositional complement.

$\{n,\ n_c\}$`-mp`: "Morphologically processed" forms of n and n_c, replacing numbers with the symbol `NUM` and names with the symbol `NAME` [à la Collins and Brooks (1995), see page 14].

These features encode the usual context for PP attachment, and have all been used in one form or another in the approaches described in Chapter 2 and elsewhere in the literature. It is worth noting that while several quadruple-based techniques—like Collins and Brooks' backed-off model (1995)—preprocess quadruples, replacing elements with lemmas or other more general forms, the variant forms here do not replace each other; lemmas, `NAME` and `NUM` symbols, and part-of-speech-tags are not considered instead of surface-form words, but rather in addition to them.

5.1.2 Structural Features

As we have seen in Chapter 2, early views of PP attachment relied on purely structural accounts of attachment preference such as the principle of right association. Techniques like the backed-off model account for the right-associative bias by, for example, favoring noun attachment if lexical evidence is equivocal. The structural features described here do not provide a simple and neat structural rule of thumb like right association. Instead, they encode various types of information about the (preliminary) parse of the sentence. They give some notion of the relation between the key head words v, n, p, and n_c and other relevant words and phrases in the sentence, as well as each other.

n-p-`distance`: A representation of the distance in number of tokens between the noun candidate and the preposition:

$$n\text{-}p\text{-}\texttt{distance} = \log_{10}\left(1 + \log_{10}(1 + d_{n-p})\right),$$

where d_{n-p} is the number of tokens between n and p.

Strictly speaking, this is not a structural feature as the purely token-based distance can be determined without any regard to a syntactic analysis of the sentence. However, the token distance does give a rough estimate of structural complexity. Also, in the case of V/N$^+$ ambiguity, it allows us to easily encode some notion of the absolute position of each noun attachment candidate, something that we were unable to easily do in our V/N$^+$ extension of the backed-off model in the previous chapter.

v-n-`path`: A representation of the syntactic relationship between the verb and noun attachment candidates in terms of the path through the parse tree between v and n. (See Figure 5.1.)

v-`subcat`: A representation of the internal structure of the verb phrase, as approximated by the labels of the VP's immediate children. In the case of a PP child, its preposition is included in the encoding. (See Figure 5.2.)

n_c-`det`: Any determiner or possessive pronoun acting as specifier of the prepositional complement, if present.

This feature allows us to distinguish, for example, the different attachment behavior effected by the possessive pronoun *my* in the following:

$$\textit{I saw the man} \begin{Bmatrix} \textit{with blue eyes} \\ \textit{with } \underline{\textit{my}} \textit{ eyes} \end{Bmatrix}.$$

n-`parent`: The label of the node immediately dominating the attachment candidate NP.

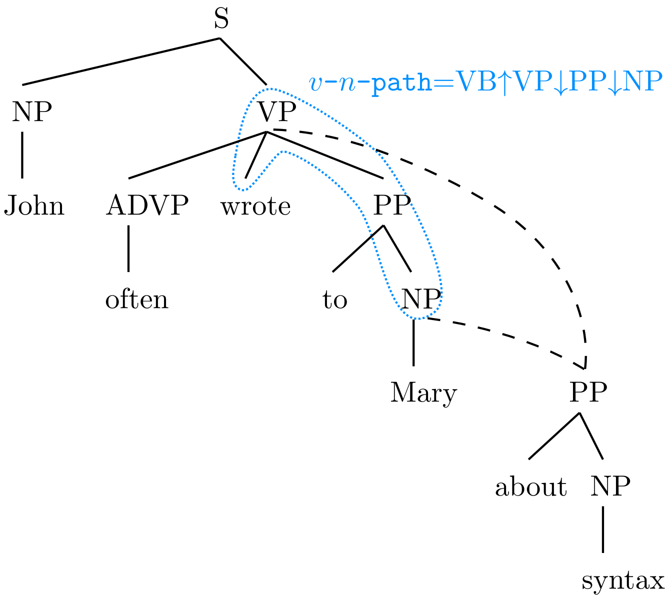


Figure 5.1: Example of v-n-`path` feature

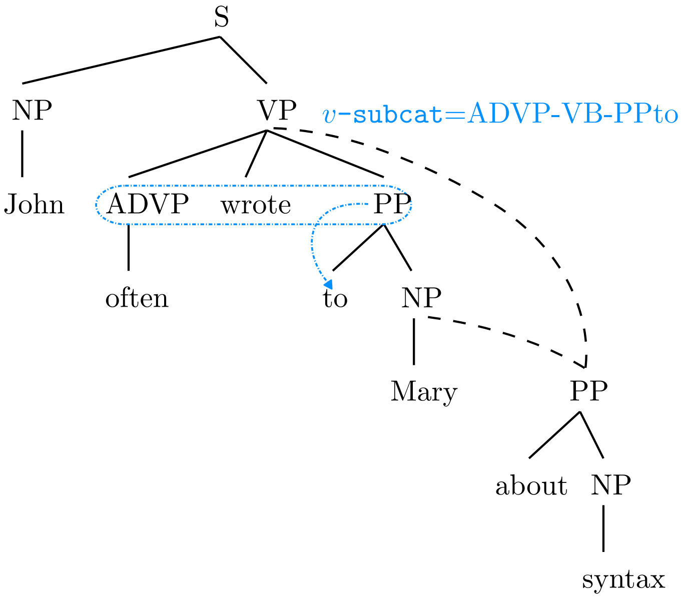


Figure 5.2: Example of v-`subcategorization` feature

n-`prep`: If n-`parent` is a PP, the preposition of that PP.

`parser-vote`: The attachment decision of the parser.

5.1.3 Semantic Features

We discussed the complementary relationship between PP attachment and semantic roles in Chapter 1. Knowing the semantic roles of PPs and their possible attachment points can be very useful in resolving attachment ambiguities. The features described in this section encode such information about the verb and noun attachment candidates, obtained from FrameNet[1] (Ruppenhofer et al., 2006).

v-`frame`: The name of the semantic frame of the verb, as listed in FrameNet. For example, the frame of the verb *wrote* in the sentence in Figures 5.1–5.2 is `Contacting`.

This type of semantic role information allows us to generalize training instances to other semantically similar cases. For example, the classifier may learn that an *about* PP is likely to attach to other verbs that evoke the `Contacting` frame, as in

$$\textit{John} \begin{Bmatrix} \textit{e-mailed} \\ \textit{called} \end{Bmatrix} \textit{Mary about syntax.}$$

n-`sr`: The semantic role of the noun attachment candidate, as listed in FrameNet. For example, the semantic role of *Mary* in the sentence in Figures 5.1–5.2 is `Addressee`.

5.1.4 Unsupervised Features

These features provide additional statistical evidence compiled from unambiguous attachment instances on the Web (see Section 2.3). Frequencies of unambiguous attachments are approximated using Google searches, and encoded in the following features:

`count-ratio`: A measure of the expected ratio of verb attachment to noun attachment, estimated from Google query hits:

$$\texttt{count-ratio} = \log_{10}\left(\frac{f(v,p,n_c)}{f(v)} \cdot \frac{f(n)}{f(n,p,n_c)}\right)$$

`pp-count`: A measure of the frequency of occurrence of the PP on the Web, estimated from Google query hits:

$$\texttt{pp-count} = \log_{10} f(p,n_c)$$

The frequencies are estimated from the following Google searches, where $f_{\text{Google}}(q)$ denotes the number of hits returned for a given query string,[2] q:

$$\begin{aligned} f(v,p,n_c) &= f_{\text{Google}}(\text{``}v\ p\ n_c\text{''}) + f_{\text{Google}}(\text{``}v\ p\ *\ n_c\text{''}) \\ &\quad + f_{\text{Google}}(\text{``}v\texttt{-lemma}\ p\ n_c\text{''}) + f_{\text{Google}}(\text{``}v\texttt{-lemma}\ p\ *\ n_c\text{''}) \end{aligned}$$

$$f(n,p,n_c) = f_{\text{Google}}(\text{``}n\ p\ n_c\text{''}) + f_{\text{Google}}(\text{``}n\ p\ *\ n_c\text{''})$$

$$f(p,n_c) = f_{\text{Google}}(\text{``}p\ n_c\text{''}) + f_{\text{Google}}(\text{``}p\ *\ n_c\text{''})$$

$$f(v) = f_{\text{Google}}(v)$$

$$f(n) = f_{\text{Google}}(n)$$

[1] `http://framenet.icsi.berkeley.edu/`

[2] These queries make use of Google's exact phrase search (quoted) and wildcard ($*$) operators.

5.2 The Medium-Context Model

A primary goal of this chapter is to reassess and build upon Olteanu and Moldovan's model with as realistic data, input, and baselines as possible, following the guidelines outlined in Section 3.3. Due to differences between their data and methodology and our own, a number of the features described in the previous section are not compatible with our experimental setup.

Olteanu and Moldovan developed and evaluated their large-context feature set in two sets of experiments over two separate corpora. The first experiment is performed over the WSJ corpus, where they extract features directly from the treebank annotations. The second experiment is performed over their own corpus of FrameNet example sentences. As these sentences are annotated with frame information, but not syntactic analysis, they extract features (other than the semantic features) from parses generated by Charniak's parser. The semantic features are only used in their FrameNet experiment, as they are extracted directly from the FrameNet annotations, which are not present in the WSJ corpus. The `parser-vote` feature is also only used in their FrameNet experiment, as no parser is used at all in their WSJ experiment. Finally, the n`-parent` feature is not used in their WSJ experiment, as it gives an indication of the correct attachment when extracted from a treebank parse.

Olteanu's FrameNet corpus is too artificial in nature to support the kind of real-world evaluation in which we are interested. While the example sentences do come from naturally occurring text, they are selected specifically to illustrate frame annotations. As such, the coverage and distribution of linguistic phenomena in this corpus may differ significantly from real text. It is thus unclear whether a truly realistic evaluation is possible using this corpus, particularly as pertains to the realism of their semantic features, as we discussed in Chapter 3. Their applicability outside of Olteanu's FrameNet corpus is questionable given that all attachment candidates in their corpus have relevant semantic information available in FrameNet, whereas only a fraction of attachment ambiguities would in reality. For these reasons, we forgo the FrameNet-based semantic features in our experiments. We do, however, experiment with incorporating semantic role information more realistically later in this chapter in Section 5.5, using an automated semantic role labeler in the place of FrameNet annotations.

We do use the same WSJ corpus[3] as Olteanu and Moldovan, but our evaluation concerns call for the use of parser input instead of directly extracting features from treebank annotations. Accordingly, we use Charniak's parser for syntactic input. While this would suggest that the `parser-vote` feature is applicable, that is not the case. Unlike Olteanu and Moldovan's FrameNet corpus, our WSJ training set coincides with the training set used by Charniak's parser. Consequently, the parser's attachment accuracy is unrealistically high (96.78%) on the training set, and would amount to an oracle during training that disappears during testing; the end result being a classifier that learns to ignore all features other than `parser-vote`.[4] Similarly, we cannot use the n`-parent` feature, as it gives an indication of

[3] Olteanu and Moldovan use the WSJ corpus from PTB-2, and partition it into training and test sets with uniform sampling. We use PTB-3 with the conventional sectional partitioning (training: 2-21, test: 23).

[4] This "disappearing oracle" effect is not an issue in Olteanu and Moldovan's experiments as they do not use these features in their WSJ experiments and their FrameNet corpus does not overlap with the parser's training data. Thus, no drastic difference in the parser's attachment decision accuracy should be expected between training and testing. However, the inclusion of these features in their FrameNet experiment is based on the conventional assumption that parsers are bad at PP attachment—or at least significantly outperformed by PP-attachment-specific techniques. As Atterer and Schütze (2007) have shown and as we

the parser's attachment decision, which again is anomalously accurate during training.

A similar effect was observed in preliminary experimentation from the feature encoding the path between the verb and noun candidates. The v-n-`path` feature leaks information about the parser's attachment decision in the same way as does the n-`parent` feature: by virtue of the fact that if n is the correct attachment site, then n-`parent` must be an NP. Therefore the path VB↑VP↓NP indicates with certainty that the PP does not attach to the noun candidate in question, while any path ending in ↓NP↓NP indicates that attachment to the given noun is likely. This result is surprising since Olteanu and Moldovan use the feature in their WSJ experiment, encoding the path through the gold-standard parse tree. Thus, unless there is some mistake in our interpretation, the results they report on the WSJ corpus are spurious, as the feature leaks information about the correct attachment from the treebank annotations.

Lastly, we are unable to use the unsupervised features as described due to changes in Google's policies, forbidding automated querying. We experimented with using cached query results (kindly provided by Marian Olteanu) to compute these features, however, since our partitionings of the WSJ corpus differ and since we extend the model to handle V/N$^+$ ambiguities, not all necessary queries are available in the cache, and the results are thus inconsistent.

The complete set of features that we use in our experiments is summarized in Figure 5.3. We refer to these as the medium-context feature set, in reference to the omission of several features from Olteanu and Moldovan's large-context feature set.

v	n	p
v-`pos`	n-`mp`	n_c
v-`lemma`	n-`pos`	n_c-`mp`
v-`subcat`	n-`lemma`	n_c-`pos`
	n-p-`distance`	n_c-`lemma`
	n-`prep`	n_c-`det`

Figure 5.3: Medium-context feature set

These features can be used with any machine learning techniques, and it is not our aim here to determine the suitability of any particular one over any other. Olteanu and Moldovan see success using support vector machines (SVMs) in their experiments, and we continue in this vein. We also observed superior performance from SVMs in our own preliminary experiments comparing several machine learning techniques using this feature set.

In all of the experiments in this chapter, we use a support vector machine from Weka[5] (Hall et al., 2009), a toolkit providing implementations of several machine learning algorithms. We use a radial basis function (RBF) kernel trained using sequential minimal optimization (SMO). The soft margin parameter, C, and the RBF kernel's inverse width parameter, γ, are optimized for each experiment using Weka's grid search functionality (using iterative 2-fold and 10-fold cross validation on the training set).

have discussed in Chapter 3, this assumption may be overly optimistic. Without an appropriate baseline evaluation of the parser's own attachment performance, we cannot be sure that features incorporating the parser's attachment decision do not yield "no-op" attachers—i.e. attachers that simply mimic the parser's decision.

[5] `http://www.cs.waikato.ac.nz/ml/weka/`

Weka's SMO classifier internally converts all discrete features into binary features, and normalizes all continuous features (in our case, distance is the only continuous feature).

5.3 Experiment 1: Medium-Context on V/N Ambiguity

Ultimately, we will use the medium-context model on V/N$^+$ ambiguities, but we first temporarily revert back to the binary case in order to better understand and properly attribute any change in performance from our new approach. In particular, we would like to distinguish how much of any performance improvement can be attributed to the switch from maximum likelihood estimation techniques to much more powerful support vector machines, and how much can be attributed to the increased context from additional features. Also, since we are unable to use all of the "optimal" feature set from Olteanu and Moldovan's experiments, we would like some notion of the relative performance of our selected subset.

We start by evaluating the performance difference between the MLE-based backed-off approach (Collins and Brooks, 1995) and an SVM-based approach. For comparability, the SVM approach is limited to using the surface form of the four head words v, n, p, and n_c, and no other features. We use the same training and test data for both techniques, extracting 4-tuples from all canonical PP attachment ambiguities in sections 2-21 of the WSJ corpus for training and section 23 for testing (yielding roughly 33 000 training tuples, and 1760 testing tuples). The gold-standard annotations are used only to determine the correct attachment, but the input seen by either system is extracted from parses automatically generated by Charniak's parser. The resulting performance is given in Table 5.1. We can see that the SVM gives slightly better performance than backed-off MLE using essentially the same features.

Table 5.1: Attachment accuracy of medium-context model (V/N ambiguity)

	PP attachment accuracy (%)
Backed-off MLE (heads)	84.97
SVM (heads only)	85.59
SVM (Medium context)	87.49
Base parser	86.50
Reranking parser	89.01

The performance of the full medium-context feature set is also given in Table 5.1, along with the corresponding attachment accuracy of Charniak's parser as a baseline. Using the additional features, the SVM performs significantly better than the backed-off model. Based on the performance of the baseline heads-only SVM, it seems the majority of this increase can be attributed to the additional contextual features, not merely the superiority of support vector machines over maximum likelihood estimation. The medium-context model also shows an improvement over Charniak's base parser, but not the reranking parser.

5.4 Experiment 2: Extending the Medium-Context Model

If the use of an SVM and the additional features of the medium-context feature set are beneficial in the case of binary ambiguity, their application in dealing with higher ambiguity cases—and the increased sparsity that comes with them—is certainly worth considering. In this section, we look at extending the medium-context model to address V/N$^+$ ambiguities.

In our analogous extension of the backed-off MLE approach discussed in Chapter 4, careful consideration of representational issues was essential, particularly with respect to backing off. In our current SVM approach, representation of instances excludes any explicit notion of backing off or the relative importance of features. The representational issues discussed in the previous section are inherent to the problem of attachment and are thus still an issue here. However, they are absorbed into the training process of the SVM. As such, from a representational perspective, extension to non-binary cases here is comparatively simple.

Features pertaining to the (sole) noun attachment candidate in the original model, are extracted for each of the possible noun attachment sites in our extended model. Specifically, our extended model includes the surface form, morphologically processed form, part of speech, and lemma of each potential noun attachment site n_i, where $1 \leq i \leq k$. Also included is a measure of distance (in tokens) between the preposition and each noun candidate. The complete feature set is summarized in Figure 5.4, with the new, adapted noun features highlighted. The attachment decision, A, for each instance is represented in the same way as in our extended back-off model: $A \in \{V, N_1, \ldots, N_i, \ldots, N_k\}$, where V denotes attachment to the verb and N_i denotes attachment to the i^{th} noun candidate. Training and classification are performed in the same way as with the binary SVM model. However, Weka transforms the multi-class classification problem into multiple pairwise classifications, as we did in our extension of the backed-off model.

v	n_i	p
v-pos	n_i-mp	n_c
v-lemma	n_i-pos	n_c-mp
v-subcat	n_i-lemma	n_c-pos
	n_i-p-distance	n_c-lemma
	n_i-prep	n_c-det

Figure 5.4: Medium-context feature set for V/N$^+$ attachment ambiguity

We evaluate this extension of the medium-context model in same manner as we did the binary model experiments above. We train an SVM with RBF kernel using sequential minimal optimization, as provided in the Weka toolkit. Again, we optimize the kernel parameters C and γ using iterative cross validation over the training set. Our data is extracted from parses of the WSJ corpus generated by Charniak's parser (section 2-21 for training and section 23 for testing) for each PP with V/N$^+$ attachment ambiguity. The performance on the WSJ corpus is given in Table 5.2, along with the usual parser baselines for comparison. The extended medium-context model significantly outperforms Charniak's base parser, but not the reranking parser.

Again, we experimented with only using the surface form of each relevant head (the same feature set as used by the extended backed-off MLE approach) to test whether the superior performance of the extended medium-context model cannot merely be attributed to the more sophisticated learning machinery. Interestingly, when using only these surface features, the SVM performs worse even than the much simpler backed-off MLE approach presented in Chapter 4, at 82.97% versus 83.73%, respectively. This may, at first, seem

Table 5.2: Attachment accuracy (%) of extended medium-context model (V/N$^+$ ambiguity)

	V/N$^+$	V/N
Heads only	82.97	85.59
Extended medium context	85.75	87.49
Base parser	84.83	86.50
Reranking parser	87.24	89.01
# of instances	2259	1758

counterintuitive given that the same comparison on the binary task showed better performance from the SVM than backed-off MLE. However, recall from Chapter 4 that careful attention to representational issues was required to extend the MLE approach to handle V/N$^+$ ambiguity. In effect, quite a bit of the intricacies and constraints of the specific problem of PP attachment—such as the importance of the preposition relative to the other head words, the uncharacteristic significance of low-count events, and the fusion of lexical, structural, semantic, etc. aspects of attachment preference—are pre-encoded in the backed-off model. In the case of the SVM, these must be learned from the training data, and it seems that a richer context than just head words is necessary to do so. In short, while additional contextual features can be beneficial in some approaches to the canonical PP attachment problem, they are absolutely essential when deciding higher ambiguity attachments.

5.5 Experiment 3: Semantic Role Labels

When introducing the problem of PP attachment in Chapter 1, we noted the complementary relationship between determining attachment and determining semantic roles. Looking at the issue of oracles and feature realism in Section 3.2.2, we recounted experiments (Mitchell, 2004; Olteanu and Moldovan, 2005) where semantic role information from manually annotated sources yielded impressive PP attachment accuracy. Here we examine the feasibility of using semantic roles, as exemplified in Example (5.3) below, in a realistic attachment task—i.e. where perfect, human-annotated semantic role labels are not available and imperfect information from an automated semantic role labeler must be used.

(5.3) *a. I flew* [$_{\textit{PP-DIRECTION}}$ *from Tokyo*] [$_{\textit{PP-DIRECTION}}$ *to New York*].
b. We have discussed the issues [$_{\textit{PP-MANNER}}$ *in detail*].

In similar fashion to Mitchell's experiments with PTB function tags (FTs), we encode the semantic role of each PP and its candidate attachment points, wherever it can be determined. Semantic role labels (SRLs) are determined using SwiRL[6] (Surdeanu and Turmo, 2005), a state-of-the-art automated semantic role labeling system. SwiRL placed among the top performing systems on the semantic role labeling shared task at CoNLL'05 (Carreras and Màrquez, 2005), achieving 80.32% precision and 72.95% recall on the WSJ test set (section 23).

To provide a point of comparison we also experiment with using function tags from the gold-standard annotations in the same fashion as did Mitchell. Note that while FTs are extracted from the gold standard in these experiments, all other input is derived from automatically generated parses. We observe a large boost in performance from the FT

[6] `http://www.surdeanu.name/mihai/swirl/`

features for both the V/N and V/N+ ambiguity cases, similar to that observed by Mitchell. In both cases, the change reflects a significant improvement over our base feature set and over Charniak's reranking parser. The automatically generated SRL features, however, do not fare nearly as well. In fact, there is barely any improvement over our base feature set. The precise accuracy figures are given in Table 5.3.

Table 5.3: Attachment accuracy (%) from semantic role features

	V/N^+	V/N
Medium context	85.75	87.49
Medium context + PTB function tags	89.33	91.01
Medium context + semantic role labels	85.79	87.94

A seemingly natural interpretation of these results might be that the current state of the art of semantic role labeling has not reached some critical threshold of performance where it can benefit attachment. While there is certainly room for improvement in semantic role labeling, how far away is this critical performance threshold from the state of the art, and is it attainable?

The complementarity of attachment and semantic role labeling is certainly worth considering here. The performance of the semantic role labeler is limited to some degree by the accuracy of its input, including attachments. If the semantic role labeler is given incorrect information about a PP's attachment, it is less likely to ascertain the correct semantic role of the PP or any related constituents. Just as PP attachment would be much more accurate with perfect semantic role information—as we see in our experiments using FTs from the gold-standard annotations—semantic role labeling would be more accurate with better PP attachment information. In our case, SwiRL obtains attachment and other information from the syntactic analysis provided by Charniak's base parser, the same analysis that we are attempting to improve. Thus, its decisions are least likely to be correct for precisely those cases for which they have the greatest potential benefit to attachment. Given the reciprocal nature of these two tasks, it may be worth exploring an iterative approach interleaving applications of attachment and semantic role labeling, where more accurate attachments are successively used as input to benefit the next iteration of semantic role labeling, and vice versa, until some convergence criterion is met.

5.6 Conclusions

In this chapter, we explored the use of additional features beyond lexical association. Our experiments here are based on the diverse feature set of Olteanu and Moldovan (2005). We first reassessed the binary model under more realistic evaluation conditions, then extended it to address attachments with V/N^+ ambiguity. In both experiments, we observed significantly better attachment accuracy from the additional features when compared to either a similarly parametrized SVM using only head-word tuples, or a state-of-the-art parser. Crucially, the results show that a richer feature set is not just beneficial, but absolutely essential when departing from the canonical V/N ambiguity task toward realistic complexity.

We have also highlighted the particular importance of realistic evaluation when exploring features. We could not use a number of Olteanu and Moldovan's features because their encoding of the parser's attachment decision—whether direct or indirect—resulted in a model that is dominated by these features, simply agreeing with whatever attachment decision is given by the parser. The inapplicability of these features would likely be missed without comparison to the baseline attachment performance of the parser. Similarly, we observed that the utility of semantic role label features is overstated when these are not applied and assessed realistically. Our experiment with automatically labeled semantic roles showed negligible improvement.

Chapter 6

Beyond Familiar Domains

Given the overwhelming dominance of lexical association as a predictor for PP attachment behavior, it should come as no surprise that shifts in vocabulary can have a profound impact on the accuracy of such predictions. We have discussed techniques, such as backing-off to a smaller, and less specific lexical context or generalizing lexical associations to semantically related words, that can approximate for terms not encountered during training. However, these techniques are much less resilient in the face of dramatic differences in term usage or large additions of new terminology between training and testing/deployment. As such, PP attachment is one aspect of parsing that is particularly susceptible to domain changes.

Naturally, if adequate resources—i.e. large labeled corpora—are available within the new domain, attachers or parsers can simply be retrained. However, manual annotation of treebanks large enough to train statistical parsers or attachers is a substantial undertaking. In many domains, the interest, personnel, and financial support needed to build large treebanks may not be there. Even where such efforts are able to proceed, annotating a sizable treebank can take several years. Should research or applications depending on accurate syntactic parses, and PP attachments in particular, simply wait?

In this chapter, we propose two different approaches to improving the PP attachment accuracy of Charniak's WSJ-trained parser in the biomedical domain. These are presented as applied solutions for improving performance on real-world parsing tasks, rather than isolated inquiries on specific forms of PP attachment ambiguity. Since the parse trees we hope to improve already specify an attachment for each and every PP, our assessment of improvement should take each of these into account. This is not to say that each technique addresses the gamut of attachment ambiguity types. Rather, wherever an approach is unable to deliberate on a particular form of attachment ambiguity, the original attachment decision of the parser prevails and is evaluated as is. We also discuss a general parsing adaptation approach (McClosky, 2010) at the end of this chapter, with an emphasis on its ability to improve PP attachment in comparison to our methods.

6.1 The Biomedical Domain

Before delving into the details of our domain adaptation experiments, we should first provide some characterization of our target domain. In particular, we outline here its significance to the field of NLP, the data we will use for our experiments, and some of the more conspicuous differences between it and the newswire data conventionally used in the training and development of parsers and attachers.

The biomedical domain is a fitting target for domain adaptation for several reasons. We are witnessing an era of tremendous growth and discovery in the biomedical sciences, where an unprecedented volume of publication demands increasing NLP support if researchers are to keep up. As a result, there is rising interest in NLP efforts addressing biomedical needs, increased need for the accurate syntactic analysis on which higher-level processing relies, and strong impetus to build new domain-targeted resources such as treebanks and lexicons. Today, there are treebanks in the domain large enough to support retraining statistical parsers. We can thus evaluate domain adaptation techniques and compare their efficacy to in-domain parsing results. However, just a few years ago, when work on two of the three approaches described in this chapter was beginning, no such resources existed.

The experiments cataloged in this chapter use the GENIA Treebank (GTB) (Tateisi et al., 2005) for evaluation.[1] The GTB contains Penn-Treebank-style syntactic annotations for each of the nearly two thousand abstracts in the GENIA corpus, a collection of molecular biology literature extracted from MEDLINE using the MeSH terms "human", "blood cells", and "transcription factors".

The text in GENIA differs from newswire text in several respects. Perhaps the most striking to those who are uninitiated in the biomedical literature, is the technical terminology. Not only do scientists write of things unknown to most outside the field, they refer to these things with long, complex, descriptive (among experts) names, which are often abbreviated to unintelligible alphanumeric sequences. Observe, for example, the following excerpts from the GENIA corpus:

(6.1) a. *At variance, in PAEC incubated with the homologous serum, NF-kappa B was strictly localized in the cell cytoplasm.*

b. *However, cyclosporin A and FK506 did not inhibit Ca2+ mobilization dependent expression of c-fos mRNA indicating that only a subset of signalling pathways regulated by Ca2+ is sensitive to these drugs.*

Here, *NF-kappa B* is an abbreviation of *nuclear factor kappa-light-chain-enhancer of activated B cells.* This term, in either form, has never been seen by the average literate human, let alone an automated parser whose only lexical exposure is a year's worth of Wall Street Journal articles. Lease and Charniak (2005) measure the unknown word rate (by token) on GENIA using a WSJ-extracted lexicon as 25.5%. In other words, a parser trained on WSJ, as was the one used in our experiments, would have no lexical knowledge of one in every four tokens in GTB—a serious detriment to a task we have shown to be so dependent on lexical information.

Higher-level differences from newswire texts are also apparent. Nominalization is rampant, with verbs often relegated to a purely functional—rather than lexical—role, as in the following:

(6.2) The inhibition of IL-2 production was observed in the CD3(+) T-lymphocyte cytoplasm as early as 4 h after activation by PMA+ionomycin.

This stylistic convention has a marked effect on PP attachment behavior. The type of attachment ambiguity shifts from the canonical V/N distinction to more ambiguities among multiple noun candidates. In a 200-abstract subset of the GTB released as an early beta version, we observe a ratio of noun to verb attachment of 2.02, nearly double the ratio of 1.15 observed in the WSJ corpus.

[1] All evaluations are performed using David McClosky's test division. Available at:
`http://bllip.cs.brown.edu/download/genia1.0-division-rel1.tar.gz`

Of course, not all aspects of biomedical texts represent increased barriers to accurate syntactic analysis. In some respects, biomedical text may be considered less linguistically complex. Taking into account that the authors of WSJ articles are writers by trade, often tasked with documenting an endless stream of familiar events—corporate mergers, changes in stock prices, introduction of new fiscal policy, etc.—somewhat more creative language use should be expected. A clever turn of phrase, figurative language, well placed witticism can make the difference between dry presentation of facts and engaging and informative prose. Conversely, the authors of scientific journal articles are researchers by trade. Their aim is to disseminate novel, and often conceptually complex, knowledge as simply as possible. This can lead to sometimes formulaic writing, particularly in the biomedical sciences, where a good deal of text is spent listing experimental parameters for the sake of reproducibility. In some ways, PP attachment may thus actually be easier in the biomedical domain.

Clegg and Shepherd (2007) benchmark the performance of several parsers in the biomedical domain. In addition to evaluating overall parser performance they look at performance on various parsing sub-tasks, including PP attachment and conjunction coordination. Interestingly, they find some of the parsers give higher PP attachment accuracy and lower conjunction accuracy relative to overall parse accuracy. Thus PP attachment may not be the most difficult aspect of parsing in biomedical texts. Perhaps coordination, as this study suggests, or some other aspect of parsing is the new bête noire of syntactic ambiguity in this domain. Even if this is the case, PP attachment is still a major difficulty, and attachment accuracy from out-of-domain resources is still low—much lower, obviously, than in the source domain of newswire texts.

6.2 Unsupervised Attachment

In Section 2.3, we introduced unsupervised attachment techniques for compiling lexical association statistics from unambiguous PP attachments, like those in Example (6.3), occurring in unlabeled text.

(6.3) The man in the park looked through the telescope.

Here we use such a technique to improve the attachment performance of a WSJ-trained parser on GENIA text.

6.2.1 Design Considerations

Several unsupervised approaches from the literature were outlined in Section 2.3, each using different strategies for detection and extraction of unambiguous attachment cases from unlabeled text. Ratnaparkhi (1998) uses heuristics (see page 22) based on the lexical category and distance of words surrounding the prepositions to determine whether or not a PP attachment is ambiguous; Volk (2001) and Olteanu and Moldovan (2005) both use Web search engines to find instances where a specific attachment site, preposition, and prepositional complement occur consecutively or in close proximity, indicating unambiguous attachment; and Kawahara and Kurohashi (2005) use heuristics based on lexical category and phrase chunks. Each approach has its trade-offs of processing time, number of unambiguous instances detected for a given quantity of raw text, and correctness of detected instances.

In our experiments, we choose to extract unambiguous instances from preliminary parses provided by Charniak's WSJ-trained parser. By using parse trees, we can detect instances where the attachment sites, preposition, and complement are not directly adjacent as in

Example (6.4) below, without resorting to approximative proximity-based pattern matching as do the approaches reviewed in Section 2.3.

(6.4) *More studies with the newer more sensitive gonadotropin assays ...*

More importantly, a preliminary syntactic analysis allows us to detect unambiguous cases much more accurately. Instead of relying on rough heuristics and word proximity, we can determine the actual attachment possibilities[2] for a given PP. An unambiguous case is then simply one where only one attachment candidate can be found. Accordingly, mistaking ambiguous PP attachments for unambiguous attachments is a much rarer occurrence.

This approach does not avoid all pitfalls, however. We do not consider the possibility of multiple verb candidates when extracting candidates. As such, PP ambiguities involving multiple verbs (V+), as in Example (6.5) may still be erroneously considered as unambiguous.

(6.5) a. *Jill bought the house Jack built with his own hands.* ⇒ $(built, with, hands)$
b. *Jill bought the house Jack built with her own money.* ⇒ $*(built, with, money)$

Also, our approach is clearly susceptible to parser errors. It may seem, at first glance, circular to rely on an out-of-domain parser for training instances to improve its own attachment accuracy. However, in detecting unambiguous attachments we disregard any PP attachment decisions made by the parser, considering only what is syntactically possible. As such, the range of parser errors having a real impact is limited to those caused by mistagging verbs as nouns (and vice versa), and the occasional incorrectly scoped coordination.

We can quantify the effect of parser errors and our mishandling of V+ ambiguities by applying the unsupervised extraction technique to treebanked data instead of unlabeled data, and evaluating the attachments of extracted triples. Doing so on GTB, we observe correct attachment in 91.72% of extracted triples. For a point of comparison, Ratnaparkhi (1998) evaluated his extraction of unambiguous attachments in a similar fashion on annotated WSJ data, reporting an accuracy of 69%.

The trade-off, of course, is that parsing is much more resource intensive than string matching or applying heuristics. As such, we were only able to use a fraction of the unlabeled corpus within the time frame of our experiments. Notwithstanding, the results from our experiments, presented in Section 6.2.3 below, suggest that additional data would not significantly increase performance.

6.2.2 Training and Classification Procedures

The training procedure, outlined in Algorithm 6.1, examines each PP in the corpus of preliminary parses, searching for instances that have only one attachment candidate. Only verb and noun attachment candidates are considered, so the resulting model consists of frequency counts of triples of the form (v, p, n_c) or (n, p, n_c).

When all PPs have been examined and unambiguous instances extracted, a second pass through the corpus is performed, counting all occurrences of verbs and nouns that were previously found to be unambiguous attachment sites. These frequencies are used during the classification stage to provide a degree of normalization of co-occurrence scores. For example, three unambiguous occurrences of $(impact, on, environment)$ in a corpus where the noun *impact* occurs only those three times are much more indicative of the lexical

[2] Procedures for detecting the attachment candidates of a PP are given in Appendix A.

association between these words than ten occurrences of $(observed, in, assay)$, where the verb *observed* occurs one hundred times, mostly with very different PP modifiers or none at all.

Algorithm 6.1 Counting unambiguous PP attachments

```
$verbAttachmentSites \leftarrow \varnothing$
$nounAttachmentSites \leftarrow \varnothing$
for all PPs do
    if $\exists v : attachmentCandidate(v, pp) \land \neg\exists n : attachmentCandidate(n, pp)$ then
        $f(v, p, n_c)$++
        $verbAttachmentSites \leftarrow verbAttachmentSites \cup \{v\}$
    else if $\neg\exists v : attachmentCandidate(v, pp) \land \exists! n : attachmentCandidate(n, pp)$ then
        $f(n, p, n_c)$++
        $nounAttachmentSites \leftarrow nounAttachmentSites \cup \{n\}$
    end if
end for
for all verbs do
    if $v \in verbAttachmentSites$ then
        $f(v)$++
    end if
end for
for all nouns do
    if $n \in nounAttachmentSites$ then
        $f(n)$++
    end if
end for
```

The resulting unsupervised model is applied to resolve V/N$^+$ ambiguous attachments—i.e. ambiguities of the form $(v, n_1, \ldots, n_i, \ldots, n_k, p, n_c)$. Each possible attachment site is scored based on the frequencies obtained from unambiguous cases. The attachment site with the highest score[3] is selected according to the following formula adapted from (Volk, 2001):

$$\operatorname*{argmax}_{x \in \{v, n_1, \ldots, n_i, \ldots, n_k\}} \frac{f(x, p, n_c)}{f(x)}.$$

In the case of ties, the lower attachment site is given precedence.

As our objective is to improve upon the performance of a parser that already provides respectably accurate attachments, even out of its domain of training, we should only override its attachment decisions when ample evidence is available from our model. Therefore, an additional constraint is added in the form of a threshold, t, where an attachment decision is made only if

$$\left[\max_{x \in \{v, n_1, \ldots, n_i, \ldots, n_k\}} \frac{f(x, p, n_c)}{f(x)} \right] > t.$$

[3] Verb and noun attachment candidates are considered equally with respect to scoring, and accordingly treated as comparable argument types in the above formula. However, to avoid conflating cases where the verb and noun forms of a word share the same spelling, separate verb and noun frequency tables are maintained. Equivalently, the attachment candidate argument, x, can be conceptualized as representing both the surface form and part of speech of the attachment site.

Where insufficient data are available and the threshold is not met, the attachment decision of the parser is left standing, as it is with ambiguities of forms other than V/N^+.

6.2.3 Results

We used the TREC Genomics 2006 corpus (Hersh et al., 2006) as unlabeled data to train unsupervised attachment models. The corpus contains 162 259 full-text articles from 49 biomedical journals distributed online through Highwire Press.[4] The first thirty thousand articles[5] (ordered by PubMed ID), containing roughly 4.84 million sentences, were parsed using Charniak's parser. From these parses, we extracted approximately 6.8 million triples. After training the unsupervised model, we used it on a development set of GTB to perform a parameter search for the optimal threshold (first over $t = 10^{-x}, 1 \leq x \leq 7$, followed by a finer-grained search over $t = x \cdot 10^{-4}, 1 \leq x \leq 9$). The optimal threshold was $t = 0.0004$.

To estimate the algorithm's learning curve and to ensure that observed behavior is not a result of insufficient data, several models were trained using variously sized subsets of the training data. Each subset was selected randomly from among the thirty thousand parsed articles. The accuracy of each of these models on the GTB test set is plotted in Figure 6.1. Performance stabilizes at around 6000 articles or more. No appreciable difference is seen when using up to five times as much training data, suggesting that additional data would not yield qualitatively different results.

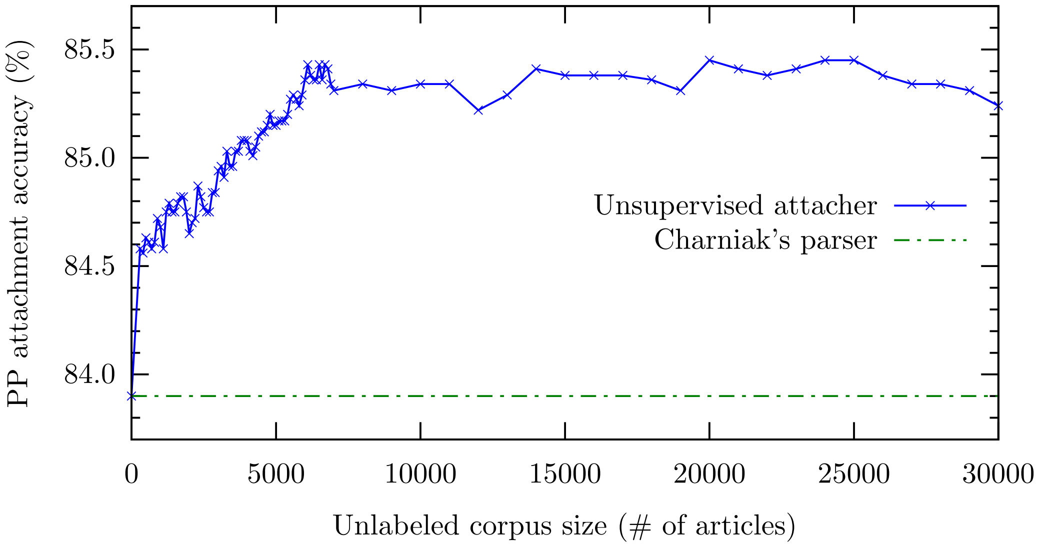


Figure 6.1: Learning curve showing attachment accuracy on GTB as a function of the number of unlabeled articles used to build the unsupervised model, contrasted with Charniak's parser as baseline

The attachment accuracy of unaltered parses from Charniak's WSJ-trained parser is also given in Figure 6.1, serving as our baseline. Even with only 100 unlabeled articles, the unsupervised method is able to improve upon this baseline. The maximum accuracy

[4] `http://www.highwire.org`

[5] A preprocessed and sentence-segmented version of the original HTML corpus was used. This version was made available to all TREC Genomics participants by fellow participant Martijn Schuemie.

of 85.45% is attained with 20 000 training articles, representing a statistically significant improvement over the parser's accuracy of 83.90%. This improvement is quite remarkable considering how cheaply it can be attained. The only additional "resource" required is a moderately sized collection of unannotated text—presumably something in ample supply for any domain in need of automated language processing. There is also certainly room for even bigger improvements. Our approach uses maximum-likelihood estimation, which may be overly simplistic for the task. In Chapter 5, we saw the MLE-based backed-off model performed worse than an SVM given the same lexical features. The performance of our unsupervised MLE approach would likely also be outperformed by more sophisticated techniques using the same unsupervised data.

There may be further potential in incorporating the unsupervised data used here within a model using more contextual features. For example, some of the features described in Chapter 5 may be usefully extractable from unambiguous attachments. Or, it may be advantageous to include unsupervised data from a new domain with such features extracted from annotated data from the source domain. Statistics over unsupervised attachment data have been successfully included in lager-context models to supplement supervised data from the same domain (Olteanu and Moldovan, 2005; Toutanova, Manning, and Ng, 2004). It may be worth investigating whether out-of-domain supervised data and in-domain unsupervised data can be beneficially combined.

Cross-domain improvement from unambiguous PP attachments is not without limitations. As previously mentioned, unambiguous training samples do not inform on all ambiguous cases, and our unsupervised training data is not perfect; we estimate roughly 8% of training triples suggest incorrect attachments, based on the accuracy of triples extracted from GTB, as described in Section 6.2.1. Still, our simple experiment here shows that unambiguous attachments can provide cheap and effective improvement in a new domain where expensive manually annotated data is not available.

6.3 Heuristic Attachment

While statistical systems may require vast quantities of high-quality annotated data to optimally adapt to domain changes, a human reader likely does not. An avid reader of the Wall Street Journal need not re-learn basic literacy skills from scratch should he or she wish to peruse the biomedical literature. In fact, likely no "retraining" is necessary at all. Consider the following sentence adapted from GTB, with esoteric terminology aplenty:

(6.6) Activation of a novel serine kinase phosphorylates c-Fos upon stimulation of T and B lymphocytes via antigen and cytokine receptors.

Even lacking the vaguest notion of what kinase, c-Fos, lymphocytes, or cytokine receptors are, or of what it means for any such things to phosphorylate each other, our Wall Street Journal reader should have little difficulty determining the attachments of each PP, and the overall syntactic structure of the sentence. Further, even rather cryptic text can become more comprehensible by reading a few more—rather than several thousand more—paragraphs. In this section, we present a domain adaptation approach where attachment behavior patterns identifiable to a human observer are encoded as heuristics.

As noted in Section 6.1, a conspicuous trait of language use in biomedical texts is the prevalence of nominalizations and a decrease in the use of verbs as true carriers of content. The result, with respect to PP attachment, is the shift from the traditional V/N ambiguity to long chains of PPs with multiple noun attachment candidates (V/N$^+$ ambiguity) and the issues that this entails, as we have already discussed in Chapter 4. In particular, we recall here the decoupling of attachment aspects discussed in that chapter. That is, when we move to predominance of multiple noun ambiguity cases with frequent nominalization, as is the case in the biomedical domain, the tidy coupling of the verb-vs-noun, structurally high-vs-low, and event-vs-entity aspects of attachment falls apart. Consider the following example:

(6.7) *a. Local correspondents* [$_{verb}$ *filed*] *reports* [$_{PP}$ *by phone*].
b. This erbA-binding site is a target for efficient [$_{noun}$ *down-regulation*] *of CAII transcription* [$_{PP}$ *by the v-erbA oncoprotein*].

Example (6.7a) contains the canonical form of PP ambiguity. The correct attachment is to *filed*, which is a verb, is the higher of the two attachment possibilities, and denotes an event. The attachment site in Example (6.7b) also denotes an event, but it is not a verb, it is neither the highest nor the lowest candidate, and its selection results in neither the fewest number of phrase nodes nor the most right-branching tree. The need for explicit event/entity distinction and the role of nominalizations therein is central to the approach presented in this section.

6.3.1 Heuristics

The heuristics described here were developed in a pilot study (Schuman and Bergler, 2006), where five articles on enzymology from PubMed Central[6] (PMC) were analyzed for PP attachment behavior. Prepositional phrases and their attachments were manually annotated, yielding 830 instances for observation, from which attachment patterns were analyzed and encoded as heuristics. These were evaluated over a further nine articles of more varied biomedical subject matter, containing an additional 3079 annotated PPs. The evaluation included an in depth analysis of where the heuristics excelled or floundered. In a later study (Schuman and Bergler, 2008), the heuristics were refined based on this analysis and based on the first beta release of the GENIA Treebank, containing 3951 PPs in 200 abstracts.

The core heuristics are based on two principles: right association (RA) and nominalization affinity (NA). We defined the former in the beginning of Chapter 2 as a preference, all things being equal, to attach new subtrees to the lowest open constituent. It is worth recalling here that while in the canonical case this principle always selects the single noun attachment option, it can be applied more subtly in the V/N$^+$ case. Essentially all the heuristics given here incorporate the RA principle in that they tend to investigate whatever cues or preferences they use to evaluate and select attachment candidates in lower candidates first, progressing to higher candidates until their heuristic criteria are met or they give up.

We introduce nominalization affinity as a principle targeted primarily at selecting between multiple noun attachment candidates. It describes a preference for attachment to event nominals rather than entities. Conceptually, this may be more accurately called event nominal affinity, or simply event affinity as preference of event attachment may be agnostic of whether the event is expressed in verbal or nominal form. However, we use the term nominalization affinity to articulate the means by which the heuristics approximate the event/entity distinction.

[6] `http://www.pubmedcentral.com`

Core Heuristics

We define three core heuristics below based on these two principles. Which of these applies to a given PP depends solely on its preposition.

Right Association

This heuristic encodes a strict application of the RA principle, selecting the lowest noun candidate irrespective of any other criteria. It is the sole heuristic for *of* and serves as the default for *for* and *from.*

Strong Nominalization Affinity

The Strong NA heuristic encodes the attachment behavior of prepositions that, in most cases, can only modify or complement events, and rarely entities. Accordingly, this heuristic selects nominalized candidates, preferring lower instances in the case where multiple candidates are nominalized. For PPs with no nominalized candidates, the verb candidate is selected for attachment.

Strong NA is applied for the prepositions *by*, *at*, *to*, *as*, *into*, *via*, *through*, *following*, *because of*, *after*, *during*, *before*, *until*, and *upon.*

Weak Nominalization Affinity

The Weak NA heuristic also encodes a preference for event attachment, but not an exclusive one. The heuristic selects the lowest nominalized candidate, if one is available. However, if no nominalization is present among the PP's attachment candidates entity attachment is not ruled out. In this case, the lowest noun candidate is selected (as with RA).

Weak NA is applied for *in*, *on*, *with*, and *without.*

Lexical & Semantic Heuristics

In addition to these core heuristics, we developed several finer-grained heuristics, encoding lexical co-occurrences, not unlike those that would be discovered by traditional statistical methods, as well as semantic relationships. Where possible, we make use of WordNet's concept hierarchy to generalize observed lexical patterns into conceptual heuristics. A fully detailed account of these finer-grained heuristics is given in Appendix B.

6.3.2 Results

These heuristics were evaluated on the same GTB test division as in the previous section. They achieved an attachment accuracy of 86.64%, a significant improvement over the 83.90% accuracy of Charniak's WSJ-trained parser on its own. The performance of the heuristics also represents a small improvement over the 85.45% accuracy of the unsupervised method of the previous section.

This improvement, however, is not achieved nearly as cheaply as that of the unsupervised method of the previous section. Rather than automatically generating a model from raw text, these heuristics required substantial human effort. Whether the cost/performance trade-off is worthwhile depends on the particularities of the overall language processing task in which these attachment methods would be applied. In some cases the performance of the unsupervised method may be sufficient, in others the small further improvement from the heuristics may be worth the extra manual work, and in still other cases attachment performance may be so crucial that incurring the even larger cost of annotating a domain-specific treebank is the optimal solution. For a dose of perspective, the time spent on developing the heuristic approach described above can be estimated at about one year of part-time work by a non-expert (in linguistics or biology), while the unsupervised approach of Section 6.2 was designed and coded in an hour or so. In contrast, the Penn Treebank represents the sustained efforts of an entire team with significant expertise spanning several years.

However, the cost in development time and the resulting overall attachment accuracy are not the only factors worth considering. These heuristics represent a fundamentally different approach to the attachment problem compared to the previous unsupervised approach, and even the various supervised techniques we have discussed throughout this thesis. There are advantageous aspects to this approach that simply cannot be measured with such metrics.

One advantage is that these heuristics draw on a diverse range of features without the need to maintain a consistent view of the attachment problem for all prepositions, or even among small subsets of PPs with the same preposition. Accordingly, different feature sets can be applied quite flexibly to very specific cases. Take for example the highly polysemous preposition *in*. In some cases, particular semantic relations between the prepositional complement and attachment candidate may strongly predict attachment behavior. In other cases, the presence of particular prenominal modifiers may better predict attachment behavior. Heuristics can narrow in on such cases and still apply lexical association, structural principles, and nominalization affinity in the more general case.

Consider also the preposition *than*. Its attachment behavior is remarkably simple, compared to other prepositions, but has almost nothing to do with the contextual features that are useful in predicting the behavior of more common prepositions. Specifically, *than* PPs attach to VPs and NPs that are modified by a comparative modifier (e.g. more, less, bigger, greener) regardless of the actual verb or noun head, or to a comparative modifier itself in the case of adjective phrases. We could train a statistical classifier, supervised or not, using a completely different attachment model for *than* PPs, or indeed a different model for each preposition. However, there may be too few instances of less common prepositions like *than* to train a separate model. Moreover, selecting the optimal feature set in each case can be no less trivial than finding appropriate heuristics, and the result may be no less brittle.

There are also advantages that are apparent when looking at a more extrinsic assessment of attachment—i.e. looking at the benefit provided to higher level processing dependent on accurate PP attachment. In Chapter 1 we presented prepositions as words that refer not to entities, actions, or properties thereof, but to relations between these. We also presented PP attachment as an essential part, along with semantic role labeling, of understanding these relations. It should thus be no surprise that PP attachment can have an important impact on information extraction tasks, particularly when identifying the participants in relations and events. Even so, PP attachment is not a standard component deployed in most information extraction pipelines. For example, Leroy, Chen, and Martinez (2003) extract relations from biomedical text using syntactic templates based heavily on prepositions, yet they forgo any PP attachment processing. Such an omission is not altogether unreasonable. Not all prepositions or PPs are equally important for a given language processing task, and not all relevant PPs are equally ambiguous. The modest increases in overall attachment accuracy that we have seen from the various approaches throughout this thesis may or may not have a big enough impact on a given information extraction task to warrant the overhead of selecting, training, and integrating a full-scale attachment component into an information extraction pipeline. The modularity of our heuristic approach can be a real advantage here. The heuristics are both comprehensible and functional on an individual basis, and can thus be easily selected and applied where needed and where they are most likely to have an impact. Kilicoglu and Bergler (2009), for example, extract biological events using dependencies provided by the Stanford Parser (Klein and Manning, 2003a). They are able to easily correct several systematic dependency errors that directly affect their patterns for identifying event participants with a few of the attachment heuristics.

Another aspect to consider when assessing the heuristic attachment approach in the context of relation or event extraction tasks is that many of the heuristic rules are specific to particular semantic relations. As a result, they indicate not only an attachment decision, but also suggest a particular event or relation and the corresponding role of the PP. This is information that simply cannot be gleaned from an approach based purely on lexical association statistics, like the backed-off model or our unsupervised adaptation in Section 6.2.

6.4 Parser Self-Training

We have looked at two quite different approaches to domain adaptation for PP attachment in the last two sections. In this section we look to the literature (McClosky and Charniak, 2008) at an approach to domain adaptation using self-training not just for PP attachment but for parsing in general. Self-training refers to retraining a model with training data it generates itself from unlabeled data. In the context of adapting a parser to a new domain, this entails using a parser trained on annotated data from the source domain to parse unlabeled text from the target domain, and then retraining the parser using these generated parses as if they had been manually annotated. This may seem like a rather counterintuitive learning strategy. Any errors in the original parser's analysis of in-domain unlabeled data are treated as correct and used for retraining, and thus reinforced in the adapted model. Previous attempts at self-training to improve parsing have led either to negligible improvement or even to decreased performance (Charniak, 1997; Steedman et al., 2003).

McClosky and Charniak (2008) apply self-training to improve the performance of Charniak's WSJ-trained reranking parser (Charniak and Johnson, 2005) in the biomedical domain. Using the original out-of-domain parser model, they parse approximately 270 000 sentences from a random selection of unannotated MEDLINE abstracts. These automatically generated parses are then added to the original WSJ training set and a new parser is trained on the combined data, with the MEDLINE sentences being weighted equally with manually annotated WSJ sentences. Evaluating this self-trained parser on GTB, they observe a 20% error reduction over the original WSJ-trained parser (by overall parse f-score).

McClosky et al. provide several analyses (2006; 2010), to better understand the benefits they attain from self-training. Unfortunately, their in-depth analyses are not performed on the MEDLINE self-training for GTB just described, but on an earlier self-training experiment. Here, they use unlabeled sentences from the North American News Text Corpus (NANC) (Graff, 1995)—which contains very similar language as WSJ—to boost performance on the WSJ corpus. As such, their analyses look at self-training as a performance enhancement within the same, or similar, domain, and are only indirectly applicable to self-training as domain adaptation. There are many factors that contribute to the improved results, but their analyses show the single most significant contribution is attributable to exposure to previously unseen head-head dependencies. This is of particular interest to our discussion, since such dependencies factor so heavily into PP attachment behavior. Surprisingly, however, their experiments suggest that PP attachment is not significantly improved by self-training.[7] We can corroborate this suggestion by directly measuring the attachment accuracy of the original parser and the NANC-self-trained parser on the WSJ corpus, in the same manner as we have evaluated the various attachment techniques throughout this thesis. Doing so, we see no noticeable difference in PP attachment accuracy.

Looking directly at the relative attachment performance between the original WSJ-trained parser and the MEDLINE-self-trained parser on GTB, however, does not give the same impression. The attachment accuracy of the MEDLINE-self-trained parser is 87.84%, a significant improvement over the 83.90% accuracy of the out-of-domain parser. Thus, contrary to the conclusion drawn from analysis of NANC self-training, we see that PP attachment can benefit substantially from self-training in the context of adaptation between two quite different domains. It would be interesting to see the same analyses given by McClosky et al. performed on the biomedical self-training data, to see if other aspects differ with the larger distance between source and target domains.

Not only do we observe a noticeable boost in PP attachment accuracy from self-training, but the improvement is also much bigger than that of either of the previously described unsupervised or heuristic adaptations. An overview of the performance of all three domain adaptation approaches is given in Table 6.1, along with the performance of the original out-of-domain parser as a baseline. The attachment accuracy of the reranking parser retrained (in the traditional, fully supervised way) on GTB is also given to provide some notion of an upper bound on adapting to this domain without labeled data.

Table 6.1: Adaptations to the biomedical domain evaluated on GTB

	PP attachment accuracy (%)
Unsupervised adaptation	85.45
Heuristic adaptation	86.64
Self-trained reranking parser (WSJ + MEDLINE)	87.84
Lower bound: Reranking parser (WSJ-trained)	83.90
Upper bound: Reranking parser (GTB-trained)	90.32

It may seem odd to end a chapter on domain adaptation for PP attachment with a general parsing adaptation—particularly when it outperforms more specialized adaptations on PP attachment accuracy. The self-trained parser gives significantly better attachment accuracy than our unsupervised adaptation from Section 6.2, without the need to narrow in on unambiguous cases, and while using an order of magnitude fewer unlabeled sentences. This may be a rather disappointing result to end on for PP attachment, but it is also a rather fitting ending. In this thesis, we have argued for attachment approaches that offer a broader, more realistic coverage using more context. Modern parsers do so, and, in the case of Charniak's reranking parser, we have seen more accurate PP attachment than any of the specialized approaches we have tested.

Perhaps looking at PP attachment in isolation from the rest of parsing is no longer necessary or beneficial. From at least one perspective that seems to be the case. Consider

[7] More accurately, McClosky et al. show that the number of prepositions in a sentence is not a factor that strongly predicts whether or not self-training will improve the f-score of that sentence. Factors that do predict improved f-score are a medium sentence length and the number of coordinating conjunctions, suggesting that resolution of conjunction ambiguity is significantly improved with self-training.

the researcher or engineer building an NLP pipeline for some application likely to benefit from highly accurate PP attachments, but not interested in advancing the state of PP attachment or parsing per se. Whether the application is in a relatively new domain as is the focus in this chapter, or otherwise, the recommended course of action would seem to be to use a highly lexicalized parser (possibly retrained or self-trained as the task requires and the data allow) and forgo any PP-attachment-specific processing unless systematic errors can be addressed with simple heuristics or some other targeted solution.

Of course, things are rarely one-dimensional and other means of syntactic analysis may be preferable for any number of reasons. The popular Stanford and Berkeley parsers (Klein and Manning, 2003a; Petrov and Klein, 2007) both feature unlexicalized or semi-lexicalized models, and their PP attachment performance can be noticeably improved with lexical-association-based PP attachment.

Chapter 7

Conclusion

Prepositional phrase attachment has long been considered one of the most difficult tasks in automated syntactic parsing of natural language text. In this thesis, we have examined several aspects of what has become the dominant view of PP attachment in NLP with an eye toward extending this view to a more realistic account of the problem. In particular, we have taken issue with the manner in which most PP attachment work is evaluated, and the degree to which traditional assumptions and simplifications no longer allow for realistically meaningful assessments. We have also argued for looking beyond the canonical subset of attachment problems, where almost all attention has been focused, toward a fuller view of the task, both in terms of the types of ambiguities addressed and the contextual information considered.

When evaluated more realistically, as we have in several different contexts throughout this thesis, it appears that state-of-the-art attachment techniques offer little advantage over a state-of-the-art parser. With an additional parse reranking stage, the parser/reranker combination performs significantly better at attachment than do attachment-specific techniques. It would seem that PP attachers are no longer a sensible component to include in NLP pipelines; if accurate PP attachments are important to a given application, an appropriate parser, like Charniak's reranking parser (Charniak and Johnson, 2005), should be used.

This is not to say that PP attachment is no longer an important topic of inquiry. We have used one parser as our baseline comparison—one that is exceptionally accurate at PP attachment. There are many other parsers based on quite different strategies that are not as accurate at PP attachment, but may excel in other areas. These can still benefit from postprocessing using existing PP attachment techniques. More importantly, there is still room for improved attachment among the best performing parsers. Perhaps future efforts should take place in a more integrated framework, like a reranker, or perhaps there is still merit in looking at attachment separately. In either case, the traditional view of PP attachment is unlikely to foster progress; a more realistic perspective is required.

The most important change required is a shift in how attachment approaches are evaluated. To start, an appropriate baseline—such as the performance of the parser whose attachments are to be improved—is essential for any realistic evaluation. Given that current parsers are capable of quite accurate attachment—some, as we have seen, even better than state-of-the-art attachers—the practical utility of an attacher simply cannot be assessed without reference to these baselines. Appropriate baselines are essential not just to validate better systems, but also to guide development. Consider some of the features discussed in Chapter 5 that encode information from the parser's (preliminary) analysis. Without a baseline comparison, it is impossible to tell whether such features contribute an additional view to the attachment model or dominate it, forcing the attachment model to parrot the parser's attachment decisions.

A realistic assessment of attachment approaches also requires that evaluation tasks bear as close a resemblance as possible to real-world tasks—where manually annotated information sources are generally not available. Atterer and Schütze (2007) argue that the conventional use of attachment quadruples extracted from manually annotated syntactic analyses is a major impediment to realistic assessment of attachment approaches, showing a large discrepancy between performance on such input as compared to that from an automated parser. While much of this discrepancy can be attributed to peculiarities of the RRR corpus (Ratnaparkhi, Reynar, and Roukos, 1994), as we have seen in Chapter 3, limiting input to what can be obtained in real-world scenarios is certainly conducive to more realistic evaluations. This is particularly important when looking at incorporating additional sources of information into an attachment model, where the difference between what can be gleaned form manually and automatically annotated sources may result in performance differences in kind rather than degree. We saw in Chapter 5 for example, that features encoding automatically labeled semantic roles yielded no benefit despite extraordinary improvements from manually annotated semantic role labels. Here, the difference is not merely that the automated semantic role labeler has less than perfect accuracy, but also that perfect semantic role labels leak information about the correct attachment.

Vital as more realistic evaluation methodology is, it will be of little benefit if we continue to focus only on the small subset of the problem that has received almost exclusive attention in the past. While binary attachment ambiguities between verb and noun candidate attachment sites represent an important and iconic subset of the general PP attachment problem, quite a bit of the problem lies outside this subset. By one account (Mitchell, 2004), verb/noun ambiguous PPs represent only 36.73% of ambiguous PPs in the Wall Street Journal corpus.

As we saw in Chapter 4, extending attachment approaches from binary V/N ambiguity to a broader range of ambiguities can be less than straightforward. Not only must we contend with potentially many more attachment possibilities, and the greater complexity and sparsity that comes with them, but the distinction between these attachment possibilities can be qualitatively quite different from the binary V/N distinction. Attachment decisions can involve lexical preferences, structural preferences, and semantic preferences, among others. In the canonical V/N ambiguity, these various dimensions of attachment preference tend to align in consistent and self-reinforcing ways—e.g. an attachment preference for a particular noun lexeme may also indicate a general preference for noun attachment over verb attachment, a structural preference for lower attachment since the noun is always structurally lower than the verb, and an entity-over-event preference since the noun usually denotes an entity while the verb generally denotes an event. Similar generalizations cannot be made when considering PPs with multiple noun attachment candidates. A lexical preference for one specific noun lexeme over another noun lexeme may or may not imply the same attachment preference if the two candidates occur in reversed order. A preference for a particular nominalized noun could imply a general preference for noun attachment over verb attachment or a preference for event attachment over entity attachment, which would generally favor verb attachment. Instead of reinforcing each other, these various dimensions of attachment can become quite convoluted and contradictory when considering attachments with higher levels of ambiguity than the traditional V/N case.

Given how different PP attachment can be when looking beyond V/N ambiguities, it should not be too surprising that the performance of a particular approach or feature set on the canonical subset may not be indicative of performance on more complete task formulations. When looking at canonical V/N ambiguous attachments, we have seen the structural principle of right association is outperformed by the backed-off model. The latter is outperformed by a support vector machine using the same head-word quadruples, which in turn is bested by an SVM using a more diverse set of features. Looking at PPs with additional noun attachment candidates (V/N^+), our extension of the backed-off model performed worse than the naive baseline of handling the additional noun candidates with right association. An SVM using head-word quadruples performed worse still. Only with the additional features was the SVM able to improve upon these approaches, as well as the parser baseline. Trying to squeeze out a bit more accuracy on canonical V/N attachments may not be helpful for overall PP attachment accuracy.

Naturally, there is no imperative requiring that the same approach, or features, be used to address all types of PP attachments uniformly; we can continue to improve on V/N attachments separately even if the resulting techniques are not applicable in the more general case. However, ignoring the rest of the PP attachment landscape unnecessarily constrains our view even of the canonical subset of the problem. The value of additional features over head-word quadruples, which improved accuracy on both V/N and V/N^+ ambiguities in our experiments in Chapter 5, is much more conspicuous when looking at the V/N^+ case. Important concepts, like the affinity of some prepositions for attachment to nominalizations, may seem practically irrelevant in the context of canonical attachment but prove quite useful in a more general context.

In a word, what we have discussed in this thesis is realism—realism in looking beyond binary V/N ambiguities at a more complete view of attachment, realism in evaluating techniques and comparing them against sensible baselines, and realism in the features used to build attachment models. If nothing else, it should be clear from our inquiry that improving PP attachment in a realistic way is no easy task. Throwing head-word quadruples at the latest machine learning technique du jour may earn additional accuracy points on the canonical task, but will likely not generate meaningful improvement across the spectrum of PPs that are currently attached quite adequately by parsers. Real progress will require a sober investigation of the feature space, and constant vigilance over the degree to which our design and evaluation decisions promote or hamper realistic assessment.

Appendix A

Extracting Attachment Candidates

Experiments throughout this thesis look at various types of ambiguity. For any given PP ambiguity, the constituent types entertained as possible attachment sites are some subset of verb phrases, noun phrases, or adjective phrases, depending on the parameters of the experiment. The eligibility for attachment site candidacy of each of these types of constituents is described below.

Verb Phrases: A maximum of one VP candidate is allowed per PP. That is the VP most closely preceding the PP, equivalently, the lowest or rightmost VP.

Adjective Phrases: ADJPs that are prenominal modifiers, as in

$[_{NP}\ [_{ADJP}$ *chemically induced*$]$ *differentiation*$]$,

are excluded from consideration, though they may be factored into a decision on the attachment suitability of the enclosing NP. Postnominal ADJPs, as in

$[_{NP}\ [_{NP}$ *mechanisms*$]$ $[_{ADJP}$ *common* $[_{PP}$ *to* $[_{NP}$ *all*$]]]]$,

or any other non-prenominal ADJPs are considered for possible attachment only if no verb attachment candidates occur between the ADJP and the PP under consideration.

Noun Phrases: NP candidates are limited to those with heads that occur before the preposition being attached and after any ADJP or VP candidates.

In all cases, only constituents within the same sentence as the PP are considered. Further, in the case where a PP occurs within one of multiple embedded sentences, only constituents within the lowest scoping sentence are eligible candidates.

Additionally, attachment candidates are excluded from consideration if their selection would introduce crossing branches into the resulting parse tree (see Figure A.1), as these are ungrammatical in English.

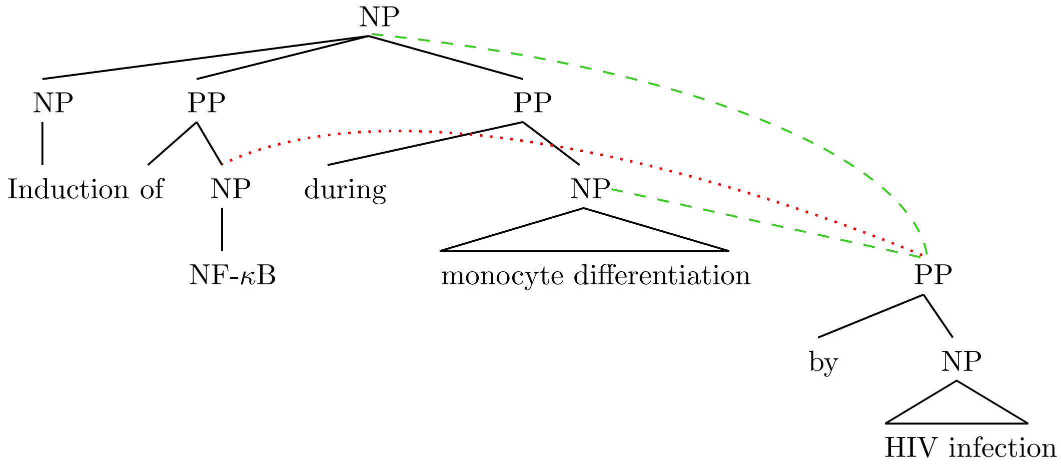


Figure A.1: An example of branch crossing—The PP cannot attach to the noun *NF-κB*, as this would result in crossed parse tree branches.

Appendix B

Attachment Heuristics

The following provides a full account of the heuristics used in the domain adaptation approach described in Section 6.3. An ordered set of heuristic rules is given for each preposition. A default set of rules is also given for prepositions that are not explicitly accounted for.

Each rule is specified using the following notation:

$$\texttt{condition} \Rightarrow \texttt{attachment},$$

where `condition` specifies the characteristics of the PP and/or its attachment candidates for which the rule applies, and `attachment` specifies the corresponding attachment decision. The latter may indicate attachment to a verb or noun candidate specified in the condition clause, or further deliberation using one of the core heuristics described in Section 6.3: Right Association, Strong Nominalization Affinity, or Weak Nominalization Affinity. Wherever more than one attachment candidate satisfies the criteria of a given rule, the candidate closest to the PP is selected for attachment.

The rules pertaining to each preposition are checked in order, and the attachment decision is determined by the first rule whose conditions are satisfied. The set of rules for each preposition is complete—i.e. all instances are guaranteed to have an applicable rule. All rulesets are also mutually exclusive—i.e. all instances are handled by one and only one ruleset, since each PP has exactly one head. (Compound heads [e.g. *because of, as of, up until*] are treated as unique prepositions, independent of their constituent parts.)

IN

1. PP complement contains a marker indicating a measure of time (e.g. *years, days*) ⇒ Strong Nominalization Affinity
2. Preposition is pre-modified by an adverb (e.g. *not in this case, only in that case*) ⇒ Strong Nominalization Affinity
3. The noun *role(s)* is a possible attachment site, and
   (a) Verb candidate is *play(s), has/have/had* ⇒ attach to the verb
   (b) Else ⇒ attach to *role(s)*
4. A noun satisfying a common lexical association is available (*increase, decrease, switch, change, shift, difference, presence, absence*) and said noun is not already modified by

a closer *in* PP
⇒ attach to the noun

5. A noun is available that is pre-modified by an adjective indicating importance (e.g. important, invaluable, essential, crucial, significant)
   ⇒ attach to the noun
6. PP complement is a hyponym of the WordNet synset *"manner, mode, style, way, fashion"* and a verb or adjective attachment candidate is available
   ⇒ attach to the verb or adjective
7. PP complement is a hyponym of the WordNet synset *"test, trial, run"* and a verb or adjective attachment candidate is available
   ⇒ attach to the verb or adjective
8. PP complement has a nominalized head
   ⇒ Strong Nominalization Affinity
9. A noun candidate is available that is a meronym of the PP complement or one of its hypernyms, as determined in WordNet
   ⇒ attach to the noun
10. Else ⇒ Weak Nominalization Affinity

FOR

1. Verb candidate is a hyponym of the WordNet synset *"want, need, require"*
   ⇒ attach to the verb
2. PP complement contains a marker indicating a measurement (e.g. *years, meters, grams*)
   ⇒ Strong Nominalization Affinity
3. Else ⇒ Right Association

FROM

1. PP complement is a hyponym of the WordNet synset *"organism, being"* and a noun candidate is available that is a hyponym of one of the WordNet synsets *"body substance"*, *"living thing, animate thing"*, or *"body part"*, and that is not already modified by a closer *from* PP
   ⇒ attach to the noun
2. A noun satisfying a common lexical association is available (*switch, change, shift, increase, decrease*) and said noun is not already modified by a closer *from* PP
   ⇒ attach to the noun
3. A noun candidate is available that is a hyponym of one of the WordNet synsets *"departure, going, going away, leaving"*, *"separation"*, or *"communication"*, and that is not already modified by a closer *from* PP
   ⇒ attach to the noun
4. Else ⇒ Right Association

TO

1. PP complement is a hyponym of the WordNet synset *"communication"* or *"communicator"* and a noun candidate is available that is a hyponym of the WordNet synset *"sensitivity, sensitiveness, sensibility"*, and that is not already modified by a closer *to* PP
   ⇒ attach to the noun

2. Verb candidate is a hyponym of the WordNet synset *"give"*
   ⇒ attach to the verb

3. A noun is available that is pre-modified by the adjective *identical*
   ⇒ attach to the noun

4. A noun is available with the root *exposure* or *response* or that ends with the suffix *ity* or *ities*, and said noun is not already modified by a closer *to* PP
   ⇒ attach to the noun

5. A noun candidate is available that is a hyponym of one of the WordNet synsets *"coupling, mating, pairing, conjugation, union, sexual union"*, *"stickiness"*, *"worth"*, *"connection, connexion, connectedness"*, *"comparison, comparing"*, *"sameness"*, *"immunity, resistance"*, *"relative, relation"*, *"way"*, *"position, spatial relation"*, *"relationship, human relationship"* , and that is not already modified by a closer *to* PP
   ⇒ attach to the noun

6. A noun satisfying a common lexical association is available (*switch, change, shift, increase, decrease*) and said noun is not already modified by a closer *to* PP
   ⇒ attach to the noun

7. Else ⇒ Strong Nominalization Affinity

AS

1. Preposition is pre-modified by the adverb *such*
   ⇒ Right Association

2. A noun is available that is pre-modified by the adjective *same*
   ⇒ attach to the noun

3. The noun *role(s)* is a possible attachment site, and
   (a) Verb candidate is *play(s), has/have/had* ⇒ attach to the verb
   (b) Else ⇒ attach to *role(s)*

4. Verb candidate satisfies a common lexical association (*use, identify, characterize*)
   ⇒ attach to the verb

5. Else ⇒ Strong Nominalization Affinity

BY

1. Prepositional complement is clausal and a verb or adjective attachment candidate is available
   ⇒ attach to the verb or adjective

2. Else ⇒ Strong Nominalization Affinity

AFTER

1. PP complement is not an NP containing a marker indicating a measure of time and a noun candidate that does indicate a measure of time is available
   ⇒ attach to the noun

2. Else ⇒ Strong Nominalization Affinity

WITH/WITHOUT

1. PP complement is clausal and a verb or adjective attachment candidate is available
   ⇒ attach to the verb or adjective

2. PP complement is a hyponym of one of the WordNet synsets *"pathological state"*, *"symptom"*, *"mental disorder"*, *"mental disturbance, disturbance, psychological disorder, folie"*, *"physiological state, physiological condition"*, or *"cardiovascular disease"* and a noun candidate is available that is a hyponym of the WordNet synset *"organism, being"*, and that is not already modified by a closer *with* PP
   ⇒ attach to the noun

3. Else ⇒ Weak Nominalization Affinity

ON

1. PP complement is a hyponym of the WordNet synset *"day of the week"*
   ⇒ Strong Nominalization Affinity

2. One of the nouns *effect*, *influence*, or *impact* is available as an attachment candidate

   (a) Verb candidate is *has/have/had* ⇒ attach to the verb

   (b) Else ⇒ attach to the noun

3. Else ⇒ Weak Nominalization Affinity

THAN

1. A noun candidate is available that is pre-modified by a comparative adjective (e.g. bigger, slower, greener)
   ⇒ attach to the noun

2. A noun candidate is available that is pre-modified by a comparative adverb (e.g. more, less)
   ⇒ attach to the noun

3. A verb or adjective attachment candidate is available
   ⇒ attach to the verb or adjective

4. Else ⇒ Do not attach (parser's attachment decision is left unmodified)

INCLUDING

1. A plural noun is available as an attachment candidate
   ⇒ attach to the noun

2. Else ⇒ Right Association

OF

Unconditionally ⇒ Right Association

AT, VIA, THROUGH, INTO, FOLLOWING, BECAUSE OF, DURING, BEFORE, UNTIL, UPON

Unconditionally ⇒ Strong Nominalization Affinity

Default

1. PP complement contains a marker indicating a measure of time
   ⇒ Strong Nominalization Affinity

2. Else ⇒ Do not attach (parser's attachment decision is left unmodified)

In addition to verb and noun attachment candidates, the heuristics also consider possible attachments to adjective phrases. For the most part, ADJP attachment candidates are considered similarly to verb candidates, and are selected as attachment sites only if they occur closer than the verb. In the case of predicative ADJPs—approximated as ADJPs immediately preceded by a copular verb—the PP is attached to the copular verb unless it can complement the adjective phrase; i.e. predicative ADJPs can take PP complements but not PP adjuncts. Predicative ADJP complements are approximated as any prepositions that co-occur with the given adjective in a development subset of GTB, except for the prepositions *of* and *than*, which are always considered complements in the context of predicative ADJPs.

Bibliography

Abney, Steven, Robert E. Schapire, and Yoram Singer. 1999. Boosting Applied to Tagging and PP Attachment. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP-VLC'99)*, volume 130, pages 132–134, College Park, MD, USA. Association for Computational Linguistics.

Altmann, Gerry and Mark Steedman. 1988. Interaction with context during human sentence processing. *Cognition*, 30(3):191–238.

Andreevskaia, Alina. 2009. *Sentence-level sentiment tagging across different domains and genres*. Ph.D. thesis, Concordia University.

Atterer, Michaela and Hinrich Schütze. 2007. Prepositional Phrase Attachment without Oracles. *Computational Linguistics*, 33(4):469–476.

Bikel, Daniel M. 2004. Intricacies of Collins' Parsing Model. *Computational Linguistics*, 30(4):479–511.

Blaheta, Don and Eugene Charniak. 2000. Assigning Function Tags to Parsed Text. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'00)*, pages 234–240, Seattle, WA, USA. Association for Computational Linguistics.

Brants, Thorsten and Alex Franz. 2006. *Web 1T 5-gram Version 1*. Linguistic Data Consortium, Philadelphia, PA, USA.

Brill, Eric and Philip Resnik. 1994. A Rule-Based Approach to Prepositional Phrase Attachment Disambiguation. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING'94)*, pages 1198–1204, Kyoto, Japan. Association for Computational Linguistics.

Carreras, Xavier and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL'05)*, pages 152–164, Ann Arbor, MI, USA. Association for Computational Linguistics.

Charniak, Eugene. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI'97)*, pages 598–603, Providence, RI, USA. Association for the Advancement of Artificial Intelligence.

Charniak, Eugene. 2000. A Maximum-Entropy-Inspired Parser. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL'00)*, pages 132–139, Seattle, WA, USA. Association for Computational Linguistics.

Charniak, Eugene and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180, Ann Arbor, MI, USA. Association for Computational Linguistics.

Church, Kenneth and Ramesh Patil. 1982. Coping with Syntactic Ambiguity or How to Put the Block in the Box on the Table. *American Journal of Computational Linguistics*, 8(3-4):139–149.

Clegg, Andrew B. and Adrian J. Shepherd. 2007. Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC Bioinformatics*, 8(1):24.

Collins, Michael. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Collins, Michael. 2000. Discriminative Reranking for Natural Language Parsing. In *Proceedings of the 17th International Conference on Machine Learning (ICML'00)*, pages 175–182, Stanford, CA, USA. Morgan Kaufmann.

Collins, Michael and James Brooks. 1995. Prepositional Phrase Attachment through a Backed-Off Model. In *Proceedings of the Third Workshop on Very Large Corpora (WVLC-3)*, pages 27–38, Cambridge, MA, USA. Association for Computational Linguistics.

Dagan, Ido, Lillian Lee, and Fernando Pereira. 1997. Similarity-based methods for word sense disambiguation. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics (EACL'97)*, pages 56–63, Madrid, Spain. Association for Computational Linguistics.

Ferreira, Fernanda and Charles Clifton, Jr. 1986. The Independence of Syntactic Processing. *Journal of Memory and Language*, 25(3):348–368.

Firth, John R. 1968. A synopsis of linguistic theory 1930-1955. In Frank R. Palmer, editor, *Selected Papers of J.R. Firth, 1952-59*. Indiana University Press.

Franz, Alexander. 1996. *Automatic Ambiguity Resolution in Natural Language Processing: An Empirical Approach*. Springer-Verlag, Secaucus, NJ, USA.

Frazier, Lyn. 1979. *On comprehending sentences: Syntactic parsing strategies*. Ph.D. thesis, University of Connecticut.

Gibson, Edward and Neal J. Pearlmutter. 1994. A Corpus-Based Analysis of Psycholinguistic Constraints on Prepositional-Phrase Attachment. In Charles Clifton, Jr., Lyn Frazier, and Keith Rayner, editors, *Perspectives on Sentence Processing*. Lawrence Erlbaum Associates, pages 181–197.

Gibson, Edward, Neal J. Pearlmutter, Enriqueta Canseco-Gonzalez, and Gregory Hickok. 1996. Recency preference in the human sentence processing mechanism. *Cognition*, 59(1):23–59.

Graff, David. 1995. *North American News Text Corpus.* Linguistic Data Consortium, Philadelphia, PA, USA.

Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18.

Harris, Zellig. 1985. Distributional structure. In Jerrold J. Katz, editor, *The Philosophy of Linguistics.* Oxford University Press, pages 26–47.

Hersh, William, Aaron M. Cohen, Phoebe Roberts, and Hari Krishna Rekapalli. 2006. TREC 2006 Genomics Track Overview. In *The Fifteenth Text Retrieval Conference (TREC'06)*, pages 52–78, Gaithersburg, MD, USA. National Institute of Standards and Technology.

Hindle, Donald. 1983. User manual for Fidditch, a deterministic parser. Naval Research Laboratory Technical Memorandum 7590–142, Naval Research Laboratory, Washington, DC, USA.

Hindle, Donald and Mats Rooth. 1993. Structural Ambiguity and Lexical Relations. *Computational Linguistics*, 19(1):103–120.

Jensen, Karen and Jean-Louis Binot. 1987. Disambiguating Prepositional Phrase Attachments by Using On-Line Dictionary Definitions. *Computational Linguistics*, 13(3-4):251–260.

Kawahara, Daisuke and Sadao Kurohashi. 2005. PP-attachment Disambiguation Boosted by a Gigantic Volume of Unambiguous Examples. In *Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP'05)*, volume 3651 of *Lecture Notes in Computer Science*, pages 188–198, Jeju Island, Korea. Springer-Verlag.

Kilicoglu, Halil and Sabine Bergler. 2009. Syntactic Dependency Based Heuristics for Biological Event Extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 119–127, Boulder, CO, USA. Association for Computational Linguistics.

Kimball, John. 1973. Seven principles of surface structure parsing in natural language. *Cognition*, 2(1):15–47.

Klein, Dan and Christopher D. Manning. 2003a. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL'03)*, pages 423–430, Sapporo, Japan. Association for Computational Linguistics.

Klein, Dan and Christopher D. Manning. 2003b. Fast Exact Inference with a Factored Model for Natural Language Parsing. In *Advances in Neural Information Processing Systems 15 (NIPS'02)*, pages 3–10, Vancouver, BC, Canada. MIT Press.

Lease, Matthew and Eugene Charniak. 2005. Parsing Biomedical Literature. In *Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP'05)*, volume 3651 of *Lecture Notes in Computer Science*, pages 58–69, Jeju Island, Korea. Springer-Verlag.

Leroy, Gondy, Hsinchun Chen, and Jesse D. Martinez. 2003. A shallow parser based on closed-class words to capture relations in biomedical text. *Journal of Biomedical Informatics*, 36(3):145–158.

Lin, Dekang and Patrick Pantel. 2001. Discovery of Inference Rules for Question Answering. *Natural Language Engineering*, 7(4):343–360.

Marcus, Mitchell P., Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):330.

McClosky, David. 2010. *Any Domain Parsing: Automatic Domain Adaptation for Parsing.* Ph.D. thesis, Brown University.

McClosky, David and Eugene Charniak. 2008. Self-Training for Biomedical Parsing. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers (ACL-HLT'08)*, pages 101–104, Columbus, OH, USA. Association for Computational Linguistics.

McClosky, David, Eugene Charniak, and Mark Johnson. 2006. Effective Self-Training for Parsing. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL'06)*, pages 152–159, New York, NY, USA. Association for Computational Linguistics.

Merlo, Paola, Matthew Crocker, and Cathy Berthouzoz. 1997. Attaching Multiple Prepositional Phrases: Generalized Backed-off Estimation. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing (EMNLP'97)*, pages 145–154, Providence, RI, USA. Association for Computational Linguistics.

Merlo, Paola and Gabriele Musillo. 2005. Accurate Function Parsing. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP'05)*, pages 620–627, Vancouver, BC, Canada. Association for Computational Linguistics.

Miller, George A. 1995. WordNet: A Lexical Database for English. *Communications of the Association for Computing Machinery*, 38(11):39–41.

Mitchell, Brian. 2004. *Prepositional Phrase Attachment using Machine Learning Algorithms.* Ph.D. thesis, University of Sheffield.

Olteanu, Marian and Dan Moldovan. 2005. PP-attachment disambiguation using large context. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT-EMNLP'05)*, pages 273–280, Vancouver, BC, Canada. Association for Computational Linguistics.

Olteanu, Marian G. 2004. Prepositional Phrase Attachment ambiguity resolution through a rich syntactic, lexical and semantic set of features applied in support vector machines learner. Master's thesis, University of Texas at Dallas.

Pedersen, Ted, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet::Similarity - Measuring the Relatedness of Concepts. In *Demonstration Papers at HLT-NAACL 2004*, pages 38–41, Boston, MA, USA. Association for Computational Linguistics.

Petrov, Slav and Dan Klein. 2007. Improved Inference for Unlexicalized Parsing. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT'07)*, pages 404–411, Rochester, NY, USA. Association for Computational Linguistics.

Quinlan, John Ross. 1986. Induction of Decision Trees. *Machine Learning*, 1(1):81–106.

Ratnaparkhi, Adwait. 1998. Statistical Models for Unsupervised Prepositional Phrase Attachment. In *Proceedings of the 17th International Conference on Computational Linguistics (COLING'98)*, pages 1079–1085, Montreal, QC, Canada. Association for Computational Linguistics.

Ratnaparkhi, Adwait, Jeff Reynar, and Salim Roukos. 1994. A maximum entropy model for prepositional phrase attachment. In *Proceedings of the Workshop on Human Language Technology*, pages 250–255, Plainsboro, NJ, USA. Association for Computational Linguistics.

Rayner, Keith, Marcia Carlson, and Lyn Frazier. 1983. The Interaction of Syntax and Semantics During Sentence Processing: Eye Movements in the Analysis of Semantically Biased Sentences. *Journal of Verbal Learning and Verbal Behavior*, 22(3):358–374.

Ruppenhofer, Josef, Michael Ellsworth, Miriam R.L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. 2006. *Framenet II: Extended Theory and Practice.* International Computer Science Institute, Berkeley, CA, USA.

Schuman, Jonathan and Sabine Bergler. 2006. Postnominal Prepositional Phrase Attachment in Proteomics. In *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, pages 82–89, New York, NY, USA. Association for Computational Linguistics.

Schuman, Jonathan and Sabine Bergler. 2008. The Role of Nominalizations in Prepositional Phrase Attachment in GENIA. In *Proceedings of the 21st Conference of the Canadian Society for Computational Studies of Intelligence (Canadian AI'08)*, volume 5032 of *Lecture Notes in Artificial Intelligence*, pages 271–282, Windsor, ON, Canada. Springer-Verlag.

Snow, Rion, Daniel Jurafsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems 17 (NIPS'04)*, pages 1297–1304, Cambridge, MA, USA. MIT Press.

Steedman, Mark, Miles Osborne, Anoop Sarkar, Stephen Clark, Rebecca Hwa, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim. 2003. Bootstrapping Statistical Parsers from Small Datasets. In *Proceedings of the 10th conference of the European Chapter of the Association for Computational Linguistics (EACL'03)*, pages 331–338, Budapest, Hungary. Association for Computational Linguistics.

Stetina, Jiri and Makoto Nagao. 1997. Corpus Based PP Attachment Ambiguity Resolution with a Semantic Dictionary. In *Proceedings of the Fifth Workshop on Very Large*

Corpora (WVLC-5), pages 66–80, Kowloon, Hong Kong. Association for Computational Linguistics.

Surdeanu, Mihai and Jordi Turmo. 2005. Semantic Role Labeling Using Complete Syntactic Analysis. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL'05)*, pages 221–224, Ann Arbor, MI, USA. Association for Computational Linguistics.

Tateisi, Yuka, Akane Yakushiji, Tomoko Ohta, and Jun'ichi Tsujii. 2005. Syntax annotation for the GENIA corpus. In *Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP'05)*, volume 3651 of *Lecture Notes in Computer Science*, pages 222–227, Jeju Island, Korea. Springer-Verlag.

Toutanova, Kristina, Christopher D. Manning, and Andrew Y. Ng. 2004. Learning Random Walk Models for Inducing Word Dependency Distributions. In *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*, Banff, AB, Canada. Association for Computing Machinery.

Volk, Martin. 2001. Exploiting the WWW as a corpus to resolve PP attachment ambiguities. In *Proceedings of Corpus Linguistics*, pages 601–606, Lancaster, England.

Voorhees, Ellen M. 1994. Query Expansion using Lexical-Semantic Relations. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 61–69, Dublin, Ireland. Springer-Verlag.

Whittemore, Greg, Kathleen Ferrara, and Hans Brunner. 1990. Empirical Study of Predictive Powers of Simple Attachment Schemes for Post-modifier Prepositional Phrases. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics (ACL'90)*, pages 23–30, Pittsburgh, PA, USA. Association for Computational Linguistics.

Zavrel, Jakub, Walter Daelemans, and Jorn Veenstra. 1997. Resolving PP attachment Ambiguities with Memory-Based Learning. In *Proceedings of the First Conference on Computational Natural Language Learning (CoNLL'97)*, pages 136–144, Madrid, Spain. Association for Computational Linguistics.

Zhao, Shaojun and Dekang Lin. 2004. 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, Hainan Island, China. Springer.