Investigations into the Use of Preposition Sense in Semantic Argument Classification

Diploma Thesis at the
Institute for Algorithms and Cognitive Systems
Cognitive Systems Lab
Prof. Dr.-Ing. Tanja Schultz, Universität Karlsruhe (TH)
Fakultät für Informatik
Universität Karlsruhe (TH)

von

cand. inform.
Daniel Hermann Richard Dahlmeier

Advisor:
Prof. Dr.-Ing. Tanja Schultz, Universität Karlsruhe (TH)
Assoc. Prof. Dr. Hwee Tou Ng, National University of Singapore

Date of Registration:	2008-03-03
Date of Submission:	2008-07-14

I declare that I have developed and written the enclosed Diploma Thesis completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, 2008-07-14

Zusammenfassung

Eine zentrale Herausforderung in der automatischen Sprachverarbeitung ist es die Bedeutung menschlicher Sprache zu erkennen. Semantic role labeling und word sense disambiguation sind zwei wichtige Hilfsmittel um eine semantische Repräsentation für einen natürlichsprachlichen Satz zu finden. Das Problem des semantic role labeling (SRL) besteht darin, semantische Rollen eines Prädikats zu erkennnen und korrekt zu klassifizieren. Das word sense disambiguation (WSD) Problem besteht darin, die korrekte Bedeutung eines Wortes in einem gegebenen Kontext zu bestimmen.

Eine Wortklasse, die sowohl häufig als auch mehrdeutig ist, sind Präpositionen. Unterscheidliche Bedeutungen von Präpositionen drücken verschiedene Beziehungen zwischen dem Komplement der Präposition und dem Rest des Satzes aus. Die Semantik der Präposition steht in Beziehung zu der semantischen Rolle der dominierenden Präpositionsphrase (PP). Um die Semantik der Präpositionsphrase zu verstehen, müsste ein System Zugang haben sowohl zu der semantischen Rolle als auch zu der Bedeutung der Präposition.

In meiner Diplomarbeit untersuche ich, in wie weit die Bedeutung von Präpositionen für die Semantic Role Labeling Aufgabe von Nutzen sein kann. Zunächst teste ich, wie genau die Bedeutung von Präpositionen mit automatischen Maschinenlernen Methoden bestimmt werden kann. Dazu annotiere ich die Präpositionen in vier Sektionen des Wall Street Journal Teils des Penn Treebank Korpus mit ihren Bedeutungen. Meine Experimente zeigen, dass ein automatisches WSD System eine grosse Menge domain-spezifischer Trainigsbeispiele benötigt, um die Bedeutung von Präpositionen akkurat klassifizieren zu könenn.

Im weiteren Verlauf der Arbeit untersuche ich verschiedene Methoden, wie die Bedeutung der Päpositionen in das SRL Problem integriert werden kann. Ich führe Experimente mit drei verschiedenen Ansätzen durch: direkte Integration als ein Feature, Kombination der Ausgaben und gemeinsame Inferenz. Meine Ergebnisse zeigen, dass die Bedeutung der Präpositionen ein nützliches Feature für SRL sein kann, aber dass die derzeitige WSD Methoden nicht präzise genug sind, um eine Verbesserung in aktuellen SRL Modellen zu erwirken.

Desweiteren entwickele ich in dieser Arbeit ein Inferenzmodell, dass das WSD und das SRL Problem für Präpositionsphrasen in einem gemeinsamen Ansatz vereint. Die Ergebnisse meiner Experimente zeigen, dass das gemeinsame Lernen von semantischen Rollen und der Bedeutung von Präposition die Klassifikationsgenauigkeit gegenüber vergleichbaren, unabhängigen Modellen für die jeweilige Aufgabe verbessert.

Abstract

A primary challenge in natural language processing (NLP) is to enable machines to disambiguate the meaning of human language. Semantic role labeling and word sense disambiguation are two key components to find a semantic representation for a sentence. Semantic role labeling is the task of determining the constituents of a sentence that represent semantic arguments with respect to a predicate an labeling each with a semantic role. Word sense disambiguation (WSD) tries to determine the correct meaning of a word in a given context.

One word class which is both frequent and highly ambiguous is preposition. The different senses of a preposition express different relations between the preposition complement and the rest of the sentence. The sense of the preposition is related to the semantic role of the domnating prepositional phrase (PP). To understand the semantics of a PP, a system would need access to both the higher level semantics of the semantic role and the finer word-token level semantics of the preposition.

In my diploma thesis, I investigate the use of preposition senses for semantic role labeling. First, I evaluate how accurate supervised machine learning methods can disambiguate prepositions for the SRL task. As part of the experiments, I manually annotate prepositions in four sections of the Wall Street Journal section of the Penn Treebank corpus. My experiments show that supervised WSD models need a large, domain specific training set to achieve accurate results.

Secondly, I investigate different methods how the preposition sense can be Incorporated into the SRL model. I conduct experiments for three different approaches: direct integration of the preposition sense as a SRL feature, classifier combination and joint inference. My results show that the preposition sense can be a useful feature for SRL, but that more accurate WSD classifiers would be needed to improve state-of-the-art SRL models. The experiments with the joint inference model show that joint learning of semantic roles and preposition senses improves the classification accuracy over competitive, individual models on each task.

Acknowledgment

I would like to express my gratitude to all those who gave me the possibility to complete this thesis.

First and foremost, I want to thank my supervisors Prof. Dr. Tanja Schultz and Prof. Dr. Ng Hwee Tou. Without their great support it would have been impossible for me to do my thesis in Singapore. I want to thank Prof. Ng for his guidance, encouraging support, and patience during the six month of my thesis. In the same way, I want to thank Prof. Schultz for her spontaneous and warm support for my idea to write the thesis in cooperation with Prof. Ng.

Special thanks go to my colleagues and co-students Liu Chang, Zhong Zhi and Wang Pidong for their assistance in implementing the experiments.

Finally, I want to thank my fiancé Yee Lin for her endless support and for always believing in me when I was struggling with my thesis.

This research is supported by a research grant R-252-000-225-112 from National University of Singapore Academic Research Fund.

Contents

List of Figures

List of Tables

1. Introduction

The core problem in computational linguistics is ambiguity. Ambiguity arises at almost every level of language processing, from word level processing tasks like part of speech tagging to high-level tasks like discourse planing. In order to understand human language, a machine has to be able to resolve these ambiguities and combine the information from different levels into an unambiguous meaning representation. Two key components to find a semantic representation for a sentence are semantic role labeling (SRL) and word sense disambiguation (WSD). Semantic role labeling is the task of determining the constituents that represent semantic arguments with respect to a predicate and labeling each with a semantic role. Semantic roles are a set of categories that group together arguments with "similar semantics". Word sense disambiguation tries to determine the correct meaning of a word in a given context. Ambiguous words occur frequently in normal English text.

One word class which is both frequent and highly ambiguous is preposition. The different senses of a preposition express different relations between the preposition complement and the rest of the sentence. The sense is related to the semantic role of the dominating prepositional phrase (PP). To understand the semantics of a PP, a system would need access to both the higher level semantics of the semantic role and the finer word-token level semantics of the preposition. In this thesis, I investigate whether the relatedness of both tasks can be exploited for SRL. To illustrate semantic roles and preposition senses, consider the following sentence:

> Daniel ate a sandwich in the morning.

A successful semantic role labeling system would yield the following information:

1. The EATER is `Daniel`.
2. The THING BEING EATEN is `a sandwich`.
3. The TIME of the event is `in the morning`.

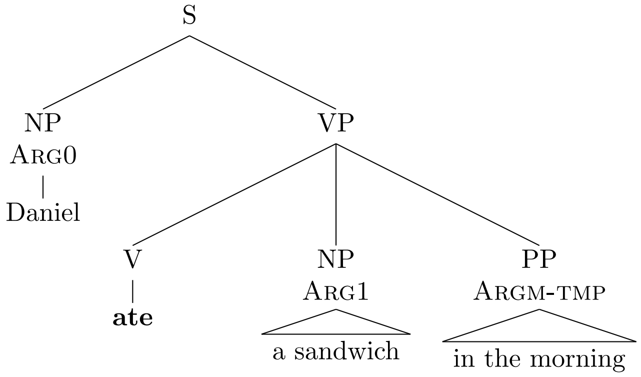
While deep semantic roles like Eater and Thing being eaten would allow a very detailed understanding of the sentence, for many NLP applications like question answering or information retrieval a more shallow approach would be sufficient. Instead of assigning highly specific roles like Eater and Thing being eaten, semantic roles can be grouped into general classes. In this example, `Daniel` would be labeled as the Agent, which is the person or entity that actively and intentionally carries out the action. The noun phrase `a sandwich` would be labeled as the Theme, which is usually the direct object of the verb and represents the entity that is mostly affected by the action. The semantic role of the prepositional phrase `in the morning` would be a temporal adjunctive argument. The labeled sentence would look like the following.

$$[\text{Daniel}]_{\text{Agent}}\ [\text{ate}]_{\text{Verb}}\ [\text{a sandwich}]_{\text{Theme}}\ [\text{in the morning.}]_{\text{Temporal}}$$

Note that the semantic roles do not change when the syntax of the sentence is altered, i.e. semantic roles are different from grammatical syntax.

$$[\text{A sandwich}]_{\text{Theme}}\text{was } [\text{eaten}]_{\text{Verb}}[\text{in the morning}]_{\text{Temporal}}\ [\text{by Daniel.}]_{\text{Agent}}$$

In recent years, the there has been a lot of interest in SRL. While early systems tried to find a semantic mapping with the help of manually created rules [Hirst1987], the focus in SRL has shifted to statistical approaches. This has become possible through the release of large, annotated text corpora which have been labeled with semantic roles by human annotators. The Propbank [Palmer et al.2005] project added a semantic layer to the Wall Street Journal section of the Penn Treebank corpus. The Penn Treebank corpus is a collection of hand-corrected, syntactic parse trees [Marcus et al.1994]. For every syntactic parse tree and every verb predicate, the arguments in the parse tree are labeled with semantic roles. Propbank does not use general labels like Agent and Theme, but simply numbers arguments with labels Arg0 to Arg5 which have verb sense specific meanings. Adjunctive arguments like time and location are labeled as Argm plus a function tag, e.g. Argm-tmp for temporal adjunctive. The syntactic parse tree with Propbank annotation for the sandwich example would look like the following:



Given a large corpus that is annotated with semantic roles, statistical methods can be applied to solve the SRL task. Most systems break the process into two subproblems:

Identification: separate the constituents that are arguments from the rest

Sense	Description	Example
1(1)	expressing the situation of something that is or appears to be enclosed or surrounded by something else	I'm living in London | dressed in their Sunday best | she saw the bus in the rear-view mirror.
2(1a)	expressing motion with the result that something ends up within or surrounded by something else	Don't put coal in the bath | he got in his car and drove off.
3(2)	expressing a period of time during which an event happens or a situation remains the case	They met in 1885 | at one o'clock in the morning | I hadn't seen him in years.
4(3)	expressing the length of time before a future event is expected to happen	I'll see you in fifteen minutes.
...	...	...

Table 1.1: Senses of the preposition `in`

Classification: assign each argument a semantic role

Both problems can be framed as classification problems. The former is a binary classification problem, the later is a multi-class classification problem where the possible classes are all possible semantic roles. By treating SRL as a classification problem, choosing an appropriate set of features becomes a key issue when designing the classifier. Previous research has identified many features that are helpful to either identify or classify semantic roles. Examples are the path from the constituent to the predicate in the parse tree, the predicate lemma or the lexical head of the constituent. However, it seems that after a few years of research, the hunt for new features has lost momentum. One contribution of this thesis is to investigate the effect of preposition senses as a potential new feature for SRL.

Prepositions are a relatively small word class, but they play an important part in English grammar. Prepositions typically appear before a noun phrase. The preposition and the noun phrase form a prepositional phrase (PP). The preposition is the *head* of the PP, the noun phrase is the *complement*. The preposition expresses the relation between the complement and another part of the sentence, for example a verb phrase or another noun phrase. In many cases the relation is temporal (`in the last month`, `on Friday`) or spatial (`in the house`, `at the bar`).

The PP `in the morning` in the sandwich sentence, for example, indicates the time of the eating event. According to the Preposition Project (TPP) dictionary[1], the preposition expresses "the period of time during which an event takes place". This particular meaning of `in` is one of the many *senses* of the preposition. Other possible senses are shown in table 1.1. The fine-grained senses in the dictionary are grouped together into a smaller number of coarse-grained senses. The coarse grained sense is given in brackets. The task of a preposition WSD system would be to automatically find the correct sense of `in` in the given context. The focus of WSD has traditionally been on nouns, verbs and adjectives, but recently WSD has been applied to prepositions as well. The SemEval 2007 evaluation exercise recently featured a competition for word sense disambiguation for prepositions [Litkowski and Hargraves2007]. The WSD task can be cast as a multi-class classification problem, where the classes for a word are all possible senses. A successful preposition WSD system would output the following assignment for the sandwich example sentence:

[1] http://www.clres.com/cgi-bin/onlineTPP/find_prep.cgi

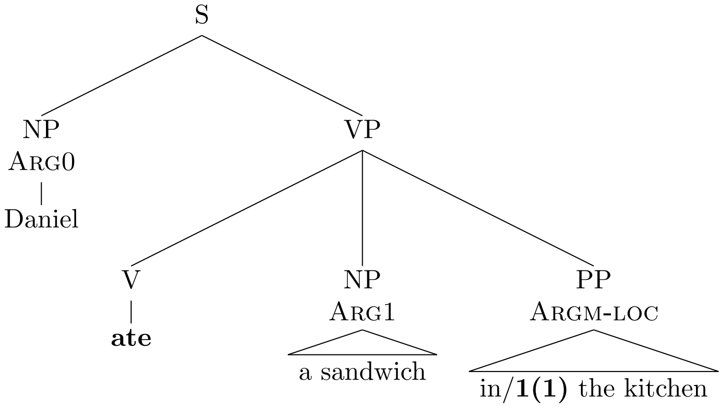
Daniel ate a sandwich in/**3(2)** the morning.

The sense of the preposition `in` and the semantic role of the prepositional phrase both seem to capture the temporal meaning of the PP.

Let us consider another example which again describes an eating event, but has a different prepositional phrase complement:

Daniel ate a sandwich in the kitchen.

In this example, the prepositional phrases describes the place where the event happened. The semantic role of the constituent and the sense of the preposition have necessarily changed. Using the Propbank semantic roles and the TPP sense definitions from above, the annotated syntactic parse tree would look like the following:



The semantic role of the PP is Argm-loc, which denotes a locative adjunctive argument. The sense of the preposition expresses the "situation of something that is or appears to be enclosed or surrounded by something else". Again, we can observe that the semantic role and the preposition sense capture similar semantic information about the PP.

1.1 Motivation

The examples in the introduction suggest that preposition senses and semantic roles of prepositional phrases are related. This is especially so for the semantic roles Argm-tmp and Argm-loc, where I expect an agreement with spatial and temporal preposition senses. The observation in this thesis is that disambiguating the preposition sense could help an SRL system to classify the semantic role of the constituent. Likewise, the semantic role of the prepositional phrase could be helpful to successfully disambiguate the sense of the preposition. To the best of my knowledge no previous research has investigated the use of preposition senses information in SRL.

1.2 Goal

The goal of the research work described in this thesis is to investigate the use of preposition sense in SRL. This implies two direct sub-problems:

1. How accurately can prepositions be automatically disambiguated for SRL?

2. How can the preposition sense be used in SRL?

The first goal of the thesis is to find an automatic tagger for preposition senses and evaluate its performance on the SRL corpus. The second goal is to use the sense to improve the existing state-of-the-art in SRL.

1.3 Contributions

The research work in this thesis makes several contributions: The first contribution is to build a preposition WSD classifier for the Propbank corpus. As part of the development process, 6,681 prepositions from the Wall Street Journal section of the Penn Treebank were manually labeled with their respective senses. This makes it possible to train a domain specific preposition WSD classifier. Secondly, I study the leverage of preposition senses as a feature for SRL classification. My experiments show that the preposition sense significantly improves the classification accuracy over a baseline model, but fails to improve a state-of-the-art SRL system. Finally, I investigate different methods to incorporate the sense in the SRL system: one classifier combination approach and one joint inference model. Instead of solving both problems sequentially, the joint inference model seeks to maximize the probability of the semantic role and the preposition sense together. To the best of my knowledge, no previous research has attempted to perform preposition WSD and SRL in an integrated approach.

1.4 Outline of the Thesis

The remainder of this thesis is organized as follows. Chapter 2 gives some background information on SRL and preposition WSD. Chapter 3 gives an overview on related work. In particular, it reviews previous work on SRL for prepositional phrases. Chapter 4 describes the methods for preposition WSD and SRL. The experiments and results can be found in chapter 5. Finally, I summarize my work in the last chapter.

2. Background

This chapter reviews some basics in computational linguistics that are necessary to understand the thesis. The first section introduces the concept of semantic roles and the SRL task. The second section describes two alternative sets of preposition senses for the Propbank corpus and the preposition WSD task.

2.1 Semantic Roles

Semantic roles [J.Fillmore1967, Gruber1970] are a linguistic concept to characterize the arguments of a predicate. Sometimes semantic roles are also referred to as *thematic roles* or *case roles* in the literature.

Compare the sandwich example sentence from the last chapter with another sentence describing the breaking of a vase.

- Daniel ate a sandwich in the morning.
- The cat broke the expensive vase.

Although the two sentences talk about different events, the subjects `Daniel` and `The cat` have something in common. Both are actively performing an action and have a direct causal responsibility for their actions. It is possible to group the two roles together into a single category. This category is called the AGENT role. Similarly, `a sandwich` and `the expensive vase` both are inanimate objects that are directly affected by the action. They fill the role of the THEME or PATIENT. Other roles include the INSTRUMENT that is used to perform the action, the FORCE or the EXPERIENCER. A list of common semantic roles with examples is given in table 2.1. Generally speaking, semantic roles capture the answers to basic Wh-Questions:

- *Who* ate the sandwich? → Daniel (AGENT)
- *What* did Daniel eat? → a sandwich (THEME)
- *When* did he eat the sandwich? → in the morning (TEMPORAL)

Semantic Role	**Description**	**Example**
AGENT	The volitional causer of an event	*The waiter* spilled the soup
EXPERIENCER	The experiencer of an event	*John* has a headache.
FORCE	The non-volitional causer of an event	*The wind* blows debris from the mall into our yards.
THEME	The participant most directly affected by an event	Only after Benjamin Franklin broke *the ice...*
RESULT	The end product of an event	The French government has built a *regulation-size baseball diamond...*
CONTENT	The preposition or content of a propositional event	Mona asked *"You met Mary Ann at a supermarket"*?
INSTRUMENT	An instrument used in an event	He turned to poaching catfish, stunning them *with a shocking device.*
BENEFICIARY	The beneficiary of an event	Whenever Ann Callahan makes hotel reservations *for her boss...*
SOURCE	The origin of the object of a transfer event	I flew in from *Boston.*
GOAL	The destination of an object of a transfer event	I drove *to Portland*

Table 2.1: Common Semantic Roles [Jurafsky and Martin2006]

SRL could provide valuable information for other NLP applications like question answering or information extraction. Semantic roles also have an application in linking theory, where they function as a mediator between the underlying meaning of a sentence and its surface realization as grammatical constituents. For example, the AGENT will usually be realized as the subject of a sentence and the THEME will be realized as the object.

An important decision when specifying a set of semantic roles is the level of granularity. Linguists have proposed a number of semantic role sets, ranging from only two meta-roles of PROTO-AGENT and PROTO-PATIENT [Foley and Valin1984, Dowty1991] to fine grained, theme specific definitions as they can be found in FrameNet [Baker et al.1998]. FrameNet is a collection of sentences from the British National Corpus (BNC) and the LDC North American Newswire corpus, which are annotated with frame-specific information. A *frame* is a structure that defines an abstract concept together with a set of semantic roles or frame elements. Take, for example, the frame KILLING. The frame describes the concept of a KILLER or CAUSE that causes the death of a VICTIM. The semantic roles KILLER, CAUSE and VICTIM are core roles of the frame. Verbs like *kill* or *assassinate* can activate the frame, but also nouns like *killer* or adjectives like *lethal.* At the time of writing, FrameNet contained over 900 frame types and over 9000 lexical units.

The problem with any set of semantic roles is that it is inherently difficult to find a good trade-off between semantic roles that are universal enough to generalize well across all predicates and at the same time are specific enough to capture valuable information. The Propbank project avoids this trade-off by defining predicate specific arguments. Instead of using the same set of semantic roles for all verbs, roles are defined specifically for each verb sense. The next section describes the details of the Propbank corpus.

2.1.1 Propbank

The Propbank project [Palmer et al.2005] created a layer of semantic annotation for syntactic parse trees from the Wall Street Journal section of the Penn Treebank II [Marcus et al.1994]. Propbank has established itself as the major corpus for SRL. The latest version of Propbank was released in September 2004. It contains approximately 113,000 annotated verb tokens for about 3,200 unique verbs, excluding auxiliary verbs like `be`, `do` or `have`. This section briefly describes the Propbank annotation scheme.

The goal of Propbank is to create a consistent annotation of predicate argument structures across different syntactic realizations of the same verb. Unlike FrameNet, which first defines semantic frames and subsequently adds example sentences, Propbank annotates all verb predicates in running text and therefore reflects the distribution of predicate argument structure in running text. For every verb predicate in the corpus, the arguments are labeled. Core arguments are labeled with consecutive numbers Arg0, Arg1,. . . , Arg5. Core arguments are verb sense specific. The verb sense is determined by the roles that can appear with it. Two verb senses are considered different, if they require different roles. This sense definition is more syntax-driven and coarser than, for example, verb senses in WordNet [Fellbaum1998]. Of the over 3,000 verbs in Propbank, only about 20% have more than one frameset and less than 100 have more than four framesets. Together with the annotated corpus, Propbank provides a lexicon that describes the meaning of core arguments for every verb sense. Each set of roles for a verb sense, together with the role definitions is called a *frameset*. A frameset includes a unique identifier for the verb sense, the verb sense meaning, a set of expected arguments together with a description and a number of examples that illustrate the use of the arguments. The sense of `break` in the sense of `break the vase`, for example, requires three core arguments: the Breaker, the Thing broken and the Result of the breaking. Note that not all arguments have to be instantiated at the same time.

break.08 sense: (cause to) not be in one piece

Arg0	:	breaker
Arg1	:	thing broken
Arg2	:	result

Example: Executives say Mr. Gorbachev's moves to break up the government's foreign trade monopoly have created uncertainties as well as opportunities.

Arg0	:	Mr. Gorbachev's
rel	:	break
Arg1	:	the government 's foreign trade monopoly

(wsj_1368.s8)

Example: Mr. Icahn has said he believes USX would be worth more if broken up into steel and energy segments.

Arg1	:	USX
rel	:	broken
Arg2	:	steel and energy segments

(wsj_0194.s26)

Argument Tag	**Description**	**Example**
Argm-dir	Directional	Workers dumped sacks *into a huge bin*
Argm-loc	Locative	The percentage of lung cancer deaths [...] be the highest for any asbestos workers studied *in Western industrialized countries.*
Argm-mnr	Manner	Men who worked *closely* with the substance.
Argm-tmp	Temporal	Four of the five surviving workers have asbestos-related diseases, including three with *recently* diagnosed cancer.
Argm-ext	Extent	PS of New Hampshire shares closed yesterday *at $3.75, off 25 cents*, in New York Stock Exchange composite trading.
Argm-rec	Reciprocal	If the stadium was such a good idea someone would build it *himself.*
Argm-prd	Secondary Predication	Pierre Vinken, 61 years old, will join the board *as a nonexecutive director* Nov.29.
Argm-pnc	Purpose	More than a few CEOs say the red-carpet treatment tempts them to return to a heartland city *for future meetings.*
Argm-cau	Cause	Five other countries will remain on that so-called priority watch list *as a result of an interim review.*
Argm-dis	Discourse	*But* for now, they 're looking forward to their winter meeting: Boca in February
Argm-adv	Adverbials	Treasures are just lying around, waiting to be picked up
Argm-mod	Modals	John does not *have* to run.
Argm-neg	Negation	John does *not* have to run

Table 2.2: Adjunctive Arguments in Propbank

In contrast, the sense of break in `break even` only requires one argument and is assigned a different frameset.

break.09 sense: not win, not lose
Arg1 : subject

Example: Federal credit programs date back to the New Deal, and were meant to break even financially.

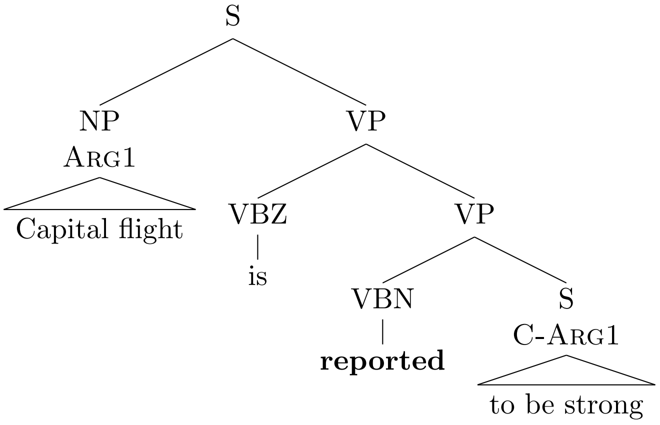
Arg0	:	Federal credit programs
rel	:	break
Argm-mnr	:	financially

(wsj_1131.s3)

Although the meaning of an argument is defined in the respective frameset, Arg0 usually denotes the Agent of the action and Arg1 denotes the tTheme. For the remaining arguments, there is no generalization.

Additional to the core arguments, a verb can have a number of adjunctive arguments that express general properties like time, location or manner of the action. These arguments do not have a verb specific meaning, but are universal to all verbs. They are labeled as Argm plus a function tag, e.g. Argm-loc for locative or Argm-tmp for temporal modifiers. Function tags are derived from the labels in the Penn Treebank. A complete list of adjunctive arguments is shown in table 2.2.

An important assumption in Propbank is that arguments align with one or more nodes in the correct parse tree. This allows to label semantic roles by annotating



(wsj_0415.s14)

Figure 2.1: Syntax tree with a discontinuous argument

nodes in the parse tree. In most cases, there is a one-to-one correspondence between arguments and nodes in the parse tree. Arguments correspond to exactly one node in the correct parse tree for 95.7% of the arguments. When the sentences are parsed with an automatic syntactic parser like Charniak's parser [Charniak2001], the arguments correspond to exactly one node for about 90% of the arguments. Still, there are a few exceptions from the one-to-one correspondence in the Propbank annotation. Sometimes arguments are discontinuous, meaning that they span more than one node. Trailing parts of the argument are labeled with a `C-` prefix. An example is shown in figure 2.1. Other arguments in Propbank are co-referential, i.e. they refer to other arguments in the tree. These arguments are labeled with a `R-` prefix. Finally, some trees in the Penn Treebank contain *null elements* that do not correspond to any word on the surface level. Examples for null elements are the implicit subject in a sentence or trace nodes that refer to another node in the tree. The challenge for computational linguistics is to build statistical models from these annotated parse trees that can automatically find the semantic roles in new, unseen instances.

2.2 Semantic Role Labeling: Task Definition

The SRL task is to determine a labeling of substrings of a sentence s with semantic roles. More formally, SRL can be defined as a mapping from the set of substrings to a label set $\mathcal{L}$ that includes all semantic roles plus a special class None for substrings that are not arguments. Every substring can be represented as a set of word indices $c \subseteq \{1, 2, \ldots, m\}$, $|s| = m$. Thus, SRL can be defined as a mapping from the powerset of word indices to the label set $\mathcal{L}$.

$$\mathrm{srl} : 2^{\{1,2,\ldots,m\}} \rightarrow \mathcal{L}$$

Because the powerset grows exponentially in the length of the sentence, SRL can be very complex. Usually the task is broken into two sub-tasks : *identification* and *classification.*

2.2.1 Identification

The task of identification is separating the substrings that are arguments from the rest. Every set of word indices is mapped to either Arg or None, indicating whether the substring is an argument or not.

$$\text{iden} : 2^{\{1,2,\ldots,m\}} \rightarrow \{\textsc{Arg}, \textsc{None}\}$$

2.2.2 Classification

The task of classification is assigning the exact argument label for all substrings that were identified as arguments during the identification step. The task is to find a mapping from substrings to semantic roles without the none label.

$$\text{clas} : \tilde{2}^{\{1,2,\ldots,m\}} \mapsto \mathcal{L} \backslash \{\textsc{none}\}$$

Where $\tilde{2}^{\{1,2,\ldots,m\}}$ denotes the subset of substrings that were marked as arguments: $\tilde{2}^{\{1,2,\ldots,m\}} = \{c \in 2^{\{1,2,\ldots,m\}} \mid \text{iden}(c) = \textsc{Arg}\}$. Note that some systems include the None label for classification as well and allow labeling a substrings as none, even if it was previously identified as an argument. This way, the classification step has the chance to correct false positive errors of the identification step.

2.2.3 SRL System Architecture

Like many problems in natural language processing, SRL is usually addressed by a pipeline of components that incrementally build the final solution. From a high-level perspective, a typical SRL pipeline consists of the following components:

1. Pre-processing
2. Local scoring
3. Global scoring

2.2.3.1 Pre-processing

The first step in SRL is to create a syntactic structure for the sentence, either a full syntactic parse or a shallow parse. Most SRL research work is done on full syntactic parses, which are either taken directly from manually corrected gold standard or automatically generated by Collin's [Collins2003] or Charniak's [Charniak2001] parser. The syntactic parse provides a structure that can subsequently be annotated with semantic roles. If arguments correspond to nodes in the (correct) parse tree, the SRL system only has to classify nodes in the parse tree, instead of all possible substrings.

It has been shown that systems that make use of full syntactic parses perform better compared to those which use shallow parses [Pradhan et al.2005]. In this thesis, I use full syntactic parses for SRL.

2.2.3.2 Local Scoring

Local scoring attempts to find the most likely semantic role for every node in the parse tree, based on information from a local context. The labels of other nodes are ignored in this step. Local scoring includes the two sub-tasks of identification and classification, so local scoring itself is divided into two steps: The *identification step*, which tries to identify the subset of nodes that are arguments to the predicate, and the *classification step* that assigns specific semantic roles. Dividing the local scoring component into two step makes training the models more efficient. During the identification step, every node in the parse tree is a potential candidate, during classification, only not-none nodes are considered. The average number of nodes per parse tree in the Propbank corpus is about 40, while the average number of not-none nodes per predicate is only 2.7. Training the classification model on the not-none nodes only, speeds up the training process. Another more subtle reason why the two steps should be addressed separately is that different features contribute unequally well to the identification and the classification step. Identification relies more on syntactic features like the path from the constituent to the predicate while classification benefits more from lexical features like the predicate lemma. Some features even improve performance in one step while decreasing performance in the other [Pradhan et al.2005].

To solve the identification step, the system has to find the probability of $y \in \{\textsc{Arg},\textsc{None}\}$, given the tree t, the predicate p and the constituent node v. Let $\Phi(\cdot,\cdot,\cdot)$ denote a feature map to an appropriate feature representation.

$$P_{iden}(y|t,p,v) = P_{iden}(y|\Phi(t,p,v)) \tag{2.1}$$

In the classification step, the system has to compute the probability of a semantic role for the constituent node, given a feature representation of its context and the result of the identification step.

$$P_{clas}(l|y,t,p,v) = P_{clas}(l|y,\Phi(t,p,v)) \tag{2.2}$$

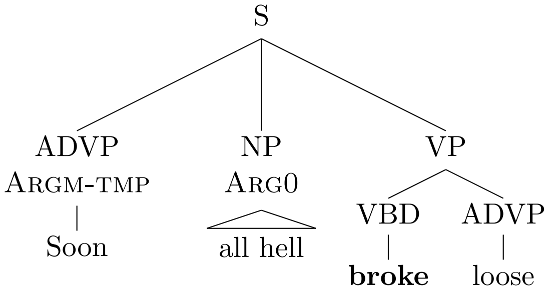
Both models can be combined into a single model by simply multiplying the probabilities. For simplicity of notation, I define a binary valued function *id* that collapses all argument labels $l \in \mathcal{L}$ except the NONE label.

$$id(l) = \begin{cases} \textsc{None} & \text{if } l = \textsc{none}, \\ \textsc{Arg} & \text{otherwise} \end{cases} \tag{2.3}$$

Local scoring is the task of finding the semantic role l that maximizes the probability of the individual role multiplied with the probability that the node is actually an argument.

$$\begin{aligned} \hat{l} &= \underset{l\in\mathcal{L}}{\operatorname{argmax}}\ P(l|t,p,v) \\ &= \underset{l\in\mathcal{L}}{\operatorname{argmax}}\ P_{iden}(id(l)|\Phi(t,p,v)) \times P_{clas}(l|id(l),\Phi(t,p,v)) \end{aligned} \tag{2.4}$$

Local scoring does not consider information from other constituent nodes. This is done in the last component, the global scoring step.



(wsj_0550.s7)

Figure 2.2: Example of ARGM-TMP and ARG0 appearing before the predicate, ARGM-TMP usually appears in first position

2.2.3.3 Global Scoring

The local scoring step maximizes the probability of the semantic role for each constituent independently. This can result in a global label sequence that is very unlikely or even impossible. Instead of finding the assignment that maximizes the probability of the individual constituents, global scoring attempts to find the assignment that maximizes the joint score of all labels, given the tree and the predicate.

$$(\widehat{l_1, \ldots, l_k}) = \operatorname*{argmax}_{(l_1,\ldots,l_k)\in\mathcal{L}^k} P(l_1, \ldots, l_k | t, p) \tag{2.5}$$

Computing the joint probability of a complete label sequence directly is not feasible, because the task is too complex. The number of possible assignments grows exponential in the length of the sentence. For a sentence of length m, there are about 20^m possible assignments, where 20 is the approximate number of possible argument labels when both core and adjunctive arguments are considered. For a normal sentence from the Wall Street Journal, this can result in several billion possible assignments. Instead of trying to solve the problem directly, global scoring takes the output of the local scoring step and re-scores it, taking into account interdependence between argument constituents.

There are two types of interdependence: *hard constraints* and *soft constraints*. Hard constraints are strict restrictions on the label sequence. For example, arguments in Propbank cannot overlap with each other or with the predicate. The following assignment would therefore be invalid, because the semantic roles ARG0 and ARG1 overlap at the word `hard`.

By [working [hard]$_{\text{ARG1}}$, he]$_{\text{ARG0}}$ **said**, you can achieve at lot.

Soft constraints are statistical tendencies for the sequence of roles and their syntactic realization. For example, there is usually not more than one temporal modifier in a sentence, or if the roles ARG0 argument and ARGM-TMP both appear before the predicate, ARGM-TMP usually appears first (see the example in figure 2.2). There are several ways to implement global scoring. One is *re-ranking*, which is a popular technique in NLP applications like parsing. The goal in re-ranking is to re-score the assigned probabilities with the help of global features, for example, the argument labels of other constituents in the parse tree or features from other nodes. The

Baseline Features [Gildea and Jurafsky2002]	
predicate	predicate lemma
path	syntactic path from constituent to predicate through the tree
phrase type	syntactic category (NP, PP, etc.) of the phrase
position	relative position to the predicate (right or left)
voice	whether the predicate is active or passive
head word	syntactic head of the phrase
sub-cat	the rule expanding the predicate node's parent

Table 2.3: Baseline Features for SRL

system in [Toutanova et al.2005] implemented a re-ranking step based on maxent models. Another global scoring method is integer linear programming (ILP). The constraints on the label sequence can be transformed into linear (in)equalities and the problem of finding the most likely label sequence that satisfies the constraints can be cast as a ILP optimization problem. The system in [Punyakanok et al.2004] uses this method. It has been shown that global scoring improves the overall SRL F1 measure by up to 3%, but because training the models for global scoring can be very expensive, not all SRL systems have a global scoring component.

The global scoring component outputs a final guess for the argument label of every constituent in the syntactic parse tree. Remember that the SRL task was defined as a labeling of substrings, not syntactic constituents. So for evaluation, the output is converted into a flat representation of argument chunks. An answer is considered correct, if the argument boundaries align with the argument boundaries in the manually annotated test sentence, and the predicted argument label is the same as the annotated argument label in the test sentence.

2.2.3.4 Features

By treating the SRL problem as a classification problem, the choice of appropriate features becomes a crucial design decision. Good features should help to discriminate between different semantic roles. Features are usually based on some insight about the preferences of semantic roles, e.g. the insight that a sentence of passive voice tends to have the theme before the agent. Features can roughly be divided into three categories:

- **Sentence level features** are shared among all arguments of the predicate. Examples are the predicate lemma, voice or predicate subcategorization.
- **Argument-specific features** are local to the constituent, for example, the phrase type or the lexical head word.
- **Argument-predicate relational features** capture information about the position of the argument inside the parse tree. An example is the path from the constituent node to the predicate in the tree.

The seven baseline features that were first proposed by [Gildea and Jurafsky2002] are shown in table 2.3. Basically all later systems make use of these baseline features. The contribution of new features is usually evaluated empirically, by either adding a feature or removing a feature from the baseline system and comparing the results of the modified system to the previous performance. A detailed study on the contribution of different features can be found in the work of [Pradhan et al.2005].

2.2.3.5 Machine Learning Algorithms

The second key issue, besides the choice of features, is the choice of the machine learning algorithm. A variety of different machine learning algorithms have been proposed for SRL. The list includes the learning algorithms SNoW [Punyakanok et al.2004], AdaBoost [Surdeanu et al.2003], memory-based learning [Tjong et al.2005], tree conditional random fields [Cohn and Blunson2005] and maximum entropy (maxent) models [Xue and Palmer2004, Toutanova et al.2005]. The dominant learning techniques in the CoNLL and Senseval competition were expectation maximization (EM) models and kernel-based methods like support vector machines (SVM). The evaluation exercises have shown that the choice of the machine learning algorithm seems to be less important, as systems with different learning algorithms achieved similar performance.

2.3 Preposition Senses

Prepositions typically appear together with a complement as part of a prepositional phrase (PP). Prepositions describe the relationship between the complement and another element of the sentence, usually a verb or noun phrase. The complement can be a noun phrase (`at the beach`) or W-ing phrase (`in public spending`). . Prepositions are a relatively small class of words, but they are among the most frequently occurring words in English. Three out of the ten most frequent English words are prepositions[1]. Because of their frequency and their importance in expressing relationships between constituents, prepositions are an important building block for English syntax and semantics.

Prepositions are highly ambiguous. One preposition can have different meanings in different contexts. Different meanings express different relationships between the prepositional phrase and the attached verb or noun phrase. Consider the following examples of the preposition `in`:

- `in` the running fiscal year
- `in` the United States

In the first example, `in` has a temporal sense and expresses the period of time during which the event happened. In the second example, `in` has a spatial sense and expresses the situation of something being enclosed in, or surrounded by something else. Human readers do not seem to have problems to understand the correct meaning of a preposition and the relationship between the prepositional phrase and the attached verb or noun phrase. For machines this meaning is less obvious. Disambiguating the preposition sense could provide valuable information about the relation between the prepositional phrases and the rest of the sentence. This motivates the problem of word sense disambiguation for prepositions. To perform WSD, we first have to find a set of suitable sense definitions for each ambiguous word. The next sub-section gives an introduction to the Preposition Project which aims to build a database of English preposition senses.

[1] http://wordcount.org/main.php

2.3.1 The Preposition Project

For open class words, there are established machine-readable dictionaries, that provide a set of sense definitions for each entry. The most popular example is WordNet [Fellbaum1998]. The Preposition Project (TPP) [Litkowski and Hargraves2005] is an attempt to create a similar lexical database for English prepositions. For each of the 334 prepositions and phrasal prepositions, the TPP database contains a set of sense definitions, a generic preposition class label and pointers to example sentences in the FrameNet corpus.

Preposition senses are defined by the TPP lexicographers based on examples of the preposition in FrameNet. The TPP lexicographers searched the FrameNet corpus for instances of the preposition and analyzed the preposition sense, considering the information from the Oxford Dictionary of English (ODE) [Soanes and Stevenson2003] and Quirk's contemporary grammar [Quirk1985]. The lexicographers note that none of these resources is complete, but that they provide complementary information about the use of the preposition. Together, they should allow the lexicographers to get a good impression of the semantics of the preposition. If the lexicographers found that the ODE definitions did not quite match the FrameNet instances or that the senses were not accurate enough, new sub-senses were created or less frequently entirely new senses. Senses are indexed with running numbers and the ODE sense of the preposition is kept in brackets as a coarse-grained sense. For example the sense *4(2)* is sense number 4 in the TPP and the ODE sense number is 2. The lexicographer further assigned a *generic preposition class* label for each sense, based on intuition. This general preposition class is not based on any formal linguistic theory, it merely gives a general characterization of sense information for the preposition, e.g. *temporal*, *spatial* ,*means/medium*, etc. The generic preposition class is very similar to the function tag of adjunctive semantic roles, in fact the generic preposition class is referred to as "semantic role label" in TPP; to avoid confusion, I stick to the term "generic class label" and reserve the term "semantic role label" for SRL. Although the function tags and generic preposition classes in TPP are similar, it is important to keep in mind that they have two different backgrounds: function tags describe the semantic role of adjunctive argument phrases with respect to a predicate. The phrase does not necessarily include a preposition. The TPP generic classes are defined for prepositions, not the predicate. The prepositional phrase might not fill a semantic role at all. Unfortunately, I found that the latest TPP release did not yet contain generic class labels for all preposition senses. For that reason, generic preposition classes are not included in the experiments in this thesis.

2.3.2 Propbank Function Tags as Preposition Sense

If a prepositional phrase in Propbank is labeled as an adjunctive argument, the semantic role of the phrase indirectly induces a "sense" for the head preposition. Consider the prepositional phrase `into a huge bin` in the following sentence:

> Workers dumped large burlap sacks of the imported material [into a huge bin], poured in cotton . . . .

The phrase is labeled with the semantic role ARGM-DIR in Propbank. This means that the prepositional phrase describes the direction of the dumping event. The function tag of the semantic role can be treated as a “sense” of the preposition `into`, i.e. the sense of `into` in this example would be DIR. This way, every prepositional phrase that is labeled as an adjunctive argument in Propbank yields one labeled instance for its head preposition. Note that this approach does not define a set of specific senses for every preposition, but uses the function tags as a general set of senses across all prepositions. Furthermore, it does not annotate all prepositions in the Propbank corpus, but only those that are the lexical head of an adjunctive argument. For prepositions that are the head of core arguments or that are not part of any semantic role, no sense information is provided.

This alternative preposition sense was first observed by [O’Hara and Wiebe2003]. They used a WSD approach to disambiguate the function tag of prepositional phrases in the Penn Treebank and the FrameNet corpus. Their work showed how semantic annotation of predicate argument structures can be used to disambiguate the sense of prepositions.

2.4 Word Sense Disambiguation for Prepositions

The task of word sense disambiguation (WSD) is to find the relevant sense of a target word, given its context. WSD has traditionally focused on open class words like nouns, verbs and adjectives, but can similarly be applied to prepositions. The first problem in WSD is to determine all possible senses of a word. The problem is not trivial, as different human readers might associate different meanings with the same word and might find it difficult to agree on the definition of its “true” senses. The problem is usually avoided by adopting the word senses from an established lexical source, e.g. a dictionary or a lexical database like TPP. But even established dictionaries might differ in their definitions for a word, so the choice of the lexical source has to be done carefully.

The second problem is to assign every word occurrence the correct sense from the set of predefined senses. There are two imaginable approaches to WSD: a *deep* and a *shallow approach*. The deep approach uses detailed knowledge about the world to infer the best fitting sense in the context, e.g. in the sentence `I live in the United States`, the preposition `in` must have a spatial sense, because the United States are a country and living in a country describes a spatial relationship between a person and a location. Unfortunately, the deep approach to WSD is not very successful in practice, mainly because such broad knowledge bases and the necessary inference methods are not available. The shallow approach to WSD does not attempt to fully understand the underlying relationship, but uses information from the sentence surface level, for example surrounding words, to find the most likely sense. Provided that sufficient labeled training data is available, WSD can be framed as a classification task and solved via supervised machine learning methods. For each preposition, a classifier is trained on a set of annotated instances and tested on another set of unseen instances.

The two major design decisions when building a WSD classifier are the choice of features and the choice of the learning algorithm. Features can roughly be categorized into two categories: *collocation features* and *co-occurrence features*. Collocation features capture position-specific information about the relation between the target word and other lexical items to the right or left of the word. Typical collocation features are surrounding words, n-grams or parts of speech. Collocation features are typically taken from a small window surrounding the target word and are effective to encode local lexical and grammatical information.

Co-occurrence features encode information of surrounding word, but ignore their exact position. Co-occurrence features give clues to the general domain of the context. The most popular example is the *bag of words* feature. A bag of words is a binary vector representation of all words appearing in a predefined window around the ambiguous word. The intuition is that a particular sense will co-occur with other words from a particular domain, e.g. when disambiguating the noun `bank`, the sense of bank as a financial institution will co-occur with other words from the financial domain like `asset`, `money`, `investment`, etc.

The feature representation of the context is the input to the machine learning algorithm. A number of different learning algorithms have been proposed for WSD, most of them showing comparable results. The list includes Naive Bayes Classifier, Decision Trees, AdaBoost and SVM. The learning algorithm creates a statistical model from the training data. During testing, the model can be used to assign the most likely sense label to new, unseen instances. In this thesis, I ask the question whether the automatically predicted preposition sense can be used to classify semantic roles.

2.5 Summary

This chapter presented an overview about semantic roles and prepositions senses. It further explained how the SRL and WSD task can be solved using automatic machine learning techniques. The next chapter reviews related work and gives more specific details on previous SRL and preposition WSD systems.

3. Related Work

This chapter reviews previous work in SRL and preposition WSD. The information in this chapter gives an overview about the state-of-the-art in both tasks.

In recent years, the NLP community has experienced a tremendous growth in interest in SRL which was accompanied by a series of evaluation exercises, including Senseval-3 [Litkowski2004] and the CoNLL 2004 and CoNLL 2005 shared task [Carreras and Màrquez2004, Carreras and Màrquez2005]. Given the vast amount of research on this topic, a complete review would be out of the scope of this chapter. Instead, the chapter points out some influential systems to give an impression of the state-of-the-art methods in SRL.

The first systems that tried to map natural language to a meaning representation were based on manually encoded rules [Hirst1987]. The problem with these knowledge-based approach is that it does not scale to larger data sets and poorly adopts to new text domains.

The statistical approach to SRL was made possible by the development of large, annotated text corpora like FrameNet, Propbank, Nombank [Meyers et al.2004] and the Chinese Propbank [Xue and Palmer2003]. With these annotated corpora and advances in domain independent machine learning it became feasible to apply statistical methods to SRL. The first statistical SRL system was presented in the groundbreaking work of [Gildea and Jurafsky2002]. They were the first to present a probabilistic model to automatically classify semantic roles. Their system has significantly influenced all later work on SRL. They proposed seven constituent dependent and constituent independent features which they combined in a backoff-based, statistical model. The model was trained in roughly 50,000 instances taken from the FrameNet corpus. Assuming correct argument boundaries, the system achieves 82% accuracy in classifying the correct semantic role. On the combined task of identification and classification, they report 65% precision and 61% recall. This sets a baseline for all following work on SRL.

Following work mainly concentrated on finding additional features and better statistical models. Practically all systems make use of the baseline features that were outlined by Gildea and Jurafsky and try improve the performance, either by finding new features, better machine learning methods or by combining the information in a more sophisticated model. The work in [Surdeanu et al.2003] reported an error reduction of over 20% on the Propbank corpus by adding a new content word feature, the POS tag of the content and head word and named entity labels. Their system was implemented using a decision tree model and achieved 89% F1 measure for identification and 83.7% accuracy for classification. The system in [Pradhan et al.2005] uses support vector machines as the underlying learning algorithm. Their results showed, that changing the learning algorithm significantly improved the performance over the baseline system. Pradhan et al. also investigated the effect of new features, such as the first and last word of the constituent or the POS and head word of siblings or the parent node. For their best system, they report a F1 measure of 93.8% for identification and 91.0% accuracy for classification on Propbank. For the combined task, their best system achieved 86.7% F1 measure. The work in [Xue and Palmer2004] showed that a careful analysis of features can lead to better results. They proposed explicit feature combinations (e.g. predicate & phrasetype) and a syntactic frame feature. Xue and Palmer used a maximum entropy model as the learning algorithm. Their results on the Propbank February 2004 release were 94% f1 measure for identification, 93% accuracy for argument classification, and 88.5% F1 measure for the combined task.

Some systems add a global scoring step after identification and classification. The need for global scoring of semantic roles arises from the fact that the semantic role of a constituent does not only depend on local features, but also on dependencies between arguments in the parse tree. These dependencies can be used in a global scoring step to find a label sequence that maximizes the probability of the semantic frame of the whole sentence while respecting the global constraints. The system in [Pradhan et al.2005] used a greedy search to find the best set of non-overlapping argument labels. [Punyakanok et al.2004] formulated constrains between semantic roles as linear (in)equalities and applied integer linear programming to find the best global assignment. They show a improvement of about 2.5% F1 measure for the combined task and about 1% for the classification task. Apart from hard constraints, there are various statistical preferences that can potentially be exploited in global scoring. The original work of [Gildea and Jurafsky2002] already proposed a smoothed relative frequency estimate of the probability of frame element multi-sets. [Pradhan et al.2005] included a trigram language model on the argument label sequence, including the predicate. They noted that the model is more helpful for core arguments than for adjunctive arguments, because adjunctives are not constrained in their position or quantity by the semantic frame. That is why the model was used for core arguments only. They reported an improvement of 0.5% F1 measure. A more complex model for global scoring was presented in the work of [Toutanova et al.2005]. They proposed a re-ranking maxent model that incorporated local features of the constituent as well as features from neighboring nodes and the global label sequence. As a result, the F1 score on the combined task improved by 2.2% compared to the local model.

Similar to open class words like nouns or verbs, prepositions can have a number of different senses. The sense expresses the relationship between the prepositional phrase and the rest of the sentence. In recent years, there have been some work on automatic word sense disambiguation of English prepositions. The work of [O'Hara and Wiebe2003] investigated preposition sense disambiguation via treebank annotation. They adopted semantic annotations from the Penn Treebank as a kind of general preposition sense. Many prepositional phrases in the Penn Treebank are annotated with function tags, e.g. PP-LOC or PP-TMP. The function tag can be interpreted as a sense label for the head preposition of the prepositional phrase. In their work, O'Hara and Wiebe trained a decision tree classifier on the seven most frequent function tags. The system was evaluated using 10-fold cross validation and achieved an average accuracy of 85.8%.

The SemEval 2007 conference featured a competition for word sense disambiguation of prepositions [Litkowski and Hargraves2007]. The task was designed as a lexical sample task, which means that the organizer provided a set of annotated example sentences for each preposition and another set of test sentences. Training and test data were taken from TPP. The best system in the competition was the MELB-YB system [Ye and Baldwin2006]. They used a maximum entropy classifier with a rich set of features, including collocation features, syntactical features and semantic role features. Their results showed that collocation features like bag of words or bag of synsets appeared to be the most effective. Syntactic features and semantic role features, on the other hand, had little positive effect. Their system achieved 69.3% fine-grained and 75.5% coarse-grained accuracy. It is not surprising that the accuracy for the SemEval 2007 Preposition WSD task is lower than O'Hara and Wiebe's result on function tag disambiguation, because the TPP sense definitions are much finer than function tags.

The work of [Dang and Palmer2005] investigated the leverage of semantic roles in WSD for verbs. They showed that semantic role features are helpful to disambiguate verb senses. Their approach is roughly the reverse of the work in this thesis, while they investigate the use of SRL features for WSD, I investigate the use of WSD features in SRL. However, they do not present results with automatically predicted semantic roles.

Considering the high accuracy reported by [O'Hara and Wiebe2003], it seems intuitive to ask whether a system for preposition sense disambiguation can be combined with a general SRL system. This approach was investigated in the work of [Ye and Baldwin2006]. Starting from O'Hara and Wiebe's work, they built a SRL system specifically for prepositional phrases. The output of the classifier was later merged with the output of a general SRL system. The task of SRL on prepositional phrases is broken down into the following sub-tasks:

1. **PP attachment:** for each preposition, determine the verb in the sentence that the prepositional phrase is attached to, or none.

2. **Semantic role classification:** for all attached prepositions, classify the semantic role. This includes core argument roles, e.g ARG0, ARG1, etc.

3. **Argument segmentation:** determine the boundaries of the semantic role.

The output of the PP attachment sub-task is the relative position of the attached verb or none, e.g -1 if the attached verb is the first verb before the prepositional phrase. Ye and Baldwin experimented with a maximum entropy classifier, a syntactic parser and a combination of both. They reported an accuracy of 86.40% over all classes and 77.48% if the none class is excluded. The second figure is more interesting, because the majority of prepositional phrases do not fill a semantic role for a verb. Instead of accuracy, precision and recall should have been reported for this step. To disambiguate the semantic role of the prepositional phrase, the same maximum entropy classifier as in the PP attachment step was used with slightly modified features. The reported accuracy was 63.36%.

To find the boundaries of the semantic role, Charniak's parser was used to create a syntax tree for the sentence. The right sibling of the preposition in the parse tree was taken as the node spanning the argument. This simple algorithm still achieved 71.48% accuracy. The results were merged with the output of the three best SRL systems from the CoNLL 2004 shared task. During merging, it of course happened that the two system did not agree on either the semantic role or the boundaries of an argument. Ye and Baldwin experimented with three simple merging strategies. Whenever there would be a conflict between the output of the prepositional phrase SRL classifier and the general SRL classifier, the system would either:

1. Only keep the answer of the general SRL classifier
2. Keep the boundary prediction of the general SRL classifier but use the semantic role predicted by they prepositional phrase SRL classifier
3. Only use the answer of the prepositional phrase SRL classifier

After merging, the final predictions are evaluated according to the standard CoNLL evaluation metric. The three best systems in CoNLL 2004 had F1 measures in the upper 60% range. For all three experiments, the overall F1 measure only increased marginally (<0.7%) and always failed to improve above the original system when a more aggressive merging strategy (strategy two or three) was chosen.

Ye and Baldwin investigated an upper bound for their system by replacing the three automatic classifiers (PP attachment, SRL classification and segmentation) with oracle functions. When all three classifiers were replaced, the performance over the baseline was increased by up to 10% F1 measure. Although this shows a large potential gain, the automatic system fails to significantly improve the performance of the general SRL system. Ye and Baldwin explain this with the more complex nature of the SRL task. Indeed, the SRL task is more difficult than just disambiguating function tags, because many prepositional phrases fill core argument roles and therefore the semantic role has a verb sense specific meaning. A system that does not have access to the governing predicate would probably not be able to learn the correct argument roles. I believe that a major shortcoming of Ye and Baldwin's system is, that the SRL classification step does not have access to the result of the previous PP attachment step, thus does not know the verb predicate that dominates the semantic frame. This certainly lowers the performance of the SRL classification step.

Finally, I need to mention the work of [Andrew et al.2004]. They propose a method to learn a joint generative inference model from partially supervised data and apply their methods to the problems of word sense disambiguation for verbs and subcategorization frames. Although they tackle different problems, their approach is similar to the joint model that I present at a later stage in this thesis. They also try to learn two related classification problems in a joint model. Yet their methods differ from mine. They use partially supervised data and a generative model, while my joint model is based on supervised data and a discriminative model. It also has to be noted that Andrew et al. do not study the effect of one task as a feature in the other task.

This chapter gave an overview about related work in SRL and preposition WSD. It summarized important work on SRL and recent work on preposition WSD, which was a major motivation to investigate prepositions in connection with SRL in this thesis. None of these systems studied the use of preposition senses for SRL, which is the main contribution of this thesis.

4. Methods

There are at least two processing steps that have to be solved to build a SRL system which incorporates preposition sense information. First, the preposition sense has to be disambiguated and second, this information must be integrated into the SRL pipeline. The methods to solve these steps are explained in this chapter. The first section describes a classifier for preposition sense disambiguation. The remaining sections explain different approaches to integrate the preposition sense into the SRL system.

4.1 Preposition Sense Disambiguation

The target corpus for SRL in this thesis is Propbank. Unfortunately, Propbank is not annotated with preposition senses. To investigate the use of the preposition sense features in SRL, the sense for the prepositions in the corpus either has to be annotated manually or the preposition sense has to be tagged by an automatic WSD classifier, before it can be integrated into the SRL system. The WSD classifier that I use in this thesis is taken from the work in [Lee and Ng2002] and was earlier implemented by Prof Ng's PhD student Zhong Zhi. The WSD system is based on support vector machines (SVM), which have shown to achieve better results than other learning algorithms. Support vector machines are a kernel-based method that transforms objects into a high-dimensional feature space and learns a classifier in that space. The SVM learning algorithm tries to construct the hyperplane that maximally separates the positive and negative examples in the training set. During testing, the classifier assigns the class for a new, unseen instance depending on what side of the hyperplane it lies on. SVM are binary classifiers. To apply SVM to a multi-class classification problem like WSD, the problem can be broken into several binary classification problems in a one-vs-all arrangement. For each class, a binary classifier is constructed. During testing, the class which receives the highest confidence value is assigned. The SVM implementation in this WSD classifier is taken from the machine learning toolkit WEKA [Witten and Frank2005]. Beside the choice of the machine learning algorithm, the most important decision when building a classifier is the choice of features. I explain the features for the WSD system in the following sub-section.

4.1.1 Features

The WSD feature design follows the work in [Lee and Ng2002] who evaluated the effect of features from different knowledge sources in combination with different learning algorithms. I chose the following three knowledge sources for the preposition sense classifier:

- Part of speech (POS) of surrounding words
- Bag of words
- Local collocations

The three knowledge sources can directly be used for preposition WSD in the same way they are used for WSD of nouns, verbs and adjectives. The details of the features are described below.

4.1.1.1 Part of Speech of Surrounding Words

The POS of surrounding words encode local grammatical information. For this knowledge source, the POS tags of surrounding tokens from a window of seven tokens around the target preposition are included as features.

$$P_{-3},\ P_{-2},\ P_{-1},\ P_0,\ P_1,\ P_2,\ P_3$$

All tokens, i.e. words or punctuation symbols, are considered, but the token must be from the same sentence as the target word. All tokens that fall outside the sentence boundaries are assigned the empty POS token *nil*. The POS tags are automatically labeled using the tagger from [Ratnaparkhi1996]. Consider the following POS-tagged sentence and the target preposition `for`:

> But/CC **for/IN** the/DT next/JJ few/JJ months/NNS ,/, these/DT boys/NNS of/IN summers/NNS long/JJ past/NN are/VBP going/VGB to/TO be/VB reveling/VBG in/IN an/DT Indian/JJ summer/NN of/IN the/DT soul/NN
> (wsj_0214.s7)

The following POS features would be extracted for this instance:

$$P_{-3}=nil,\ P_{-2}=nil,\ P_{-1}=\text{CC},\ P_0=\text{IN},\ P_1=\text{DT},\ P_2=\text{JJ},\ P_3=\text{JJ},$$

4.1.1.2 Bag of Words

This knowledge source encodes co-occurrence information from other words from a large window around the target word. I follow the configuration of the SemEval 2007 preposition WSD task, where one sentence is given as context for each instance. Thus, I consider all words from the same sentence for this knowledge source. The input sentence is tokenized and all tokens that do not contain at least one alphabet character, such as punctuation symbols and numbers, and all words that appear on a stopword list are removed. The remaining words are converted to lower case and replaced by their morphological root form using the stemming mechanism from WordNet. Every unique, stemmed word contributes one binary feature, indicating whether the word is present in the context or not. The position of the word in the sentence is ignored. Let the set of observed words in the training data be: {`boat`, `boy`, `india`, `universe`, `summer`}

> But **for** the next few months , these boys of summers long past are going to be reveling in an Indian summer of the soul .

The bag of binary features for the example sentence would be $\{0, 1, 1, 0, 1\}$.

4.1.1.3 Local Collocations

Local collocations encode position-specific lexical information from words within a small window around the target word. For this knowledge source, the WSD classifier extracts unigrams, bigrams and trigrams from a window of seven tokens around the target word. Words are converted to lower case, but no stemming and removal of stopwords, numbers or punctuation symbols is performed. In total, this yields 11 collocation features:

$$C_{-1-1},\ C_{11},\ C_{-2-2},\ C_{22},\ C_{-2-1},\ C_{-11},\ C_{12},\ C_{-3-1},\ C_{-21},\ C_{-12},\ C_{13}$$

The symbol C_{ij} denotes the collocation starting at offset i and ending at offset j. A negative offset denotes a token to the left of the target word, a positive offset denotes a token to the right. The target word itself is not included, but its position inside the n-gram is marked with a special character ‘_’. Every seen n-gram value from the training set is one possible feature value for this knowledge source. Similar to surrounding POS, collocation features do not cross sentence boundaries. If a token falls outside the sentence, it is replaced by the empty token symbol *nil*.

The values of the collocations C_{-21} and C_{-12} in the example

> But **for** the next few months, these boys of summer. . .

would be C_{-21}=*nil_but_._the* and C_{-12}=*but_._the_next*.

4.1.2 Predicting Preposition Sense for Propbank

The WSD classifier can be used to automatically label prepositions in Propbank with TPP preposition senses. To ensure that the learning algorithm and features are well chosen, the classifier was tested on the SemEval 2007 preposition WSD task, using the official training and test data as provided by the organizer. The classifier achieved 71.1% fine-grained and 77.1% coarse-grained accuracy on the official test set, which is better than the best participating system in the competition, which achieved 69.3% and 75.5% accuracy. The results show that the my adapted WSD classifier achieves state-of-the-art results for preposition WSD.

The classifier that was trained on the SemEval training data is used to automatically tag prepositions in Propbank with TPP sense labels. The problem is that the accuracy of the prediction cannot be measured, because of the lack of annotated instances that can be used as a test set. It is well known that the performance of a classifier drops when it is applied to instances from a different domain and the accuracy of classifier will most likely be lower compared to the SemEval test set. To find out how accurate the automatically predicted sense labels are, I manually annotated the seven most frequent prepositions in a part of the Propbank corpus with TPP preposition senses. According to [Jurafsky and Martin2006], the seven most

Preposition	Total	Training (sec 2-4)	Test (sec 23)
at	514	345	169
for	766	546	220
in	2094	1499	595
of	1525	1205	320
on	547	361	186
to	752	483	269
with	483	328	155
Total	6681	4767	1914

Table 4.1: Number of manually annotated prepositions in Propbank

frequent prepositions in English are `at`, `for`, `in`, `of`, `on`, `to` and `with` (in alphabetical order). The annotation was performed in the following manner: first, all sentences which have one of the above prepositions as the lexical head of a prepositional phrase were automatically extracted. The position of the preposition was marked in the sentence. By only considering prepositional phrases, occurrences of the word `to` before infinitives and instances of particle usage of prepositions, such as phrasal verbs (e.g. `So I went` *`on`* `for some days ...`) were automatically excluded. Idiomatic usage of prepositions, like `for example` or `in fact` and complex preposition constructions that involve more than one word like `because of` or `instead of` were manually excluded and compiled into a stoplist.

During the annotation process, the annotator was displayed the context sentence with the marked preposition, together with the guess of the automatic WSD classifier that was trained on SemEval data and a list of possible senses for the preposition. The annotator then had to decide to either keep the predicted sense, assign a different sense or skip to the next instance. The online TPP dictionary and the official SemEval 2007 training data were used as reference to ensure that the annotation tallies with the annotation of the SemEval organizers. Using this semi-automatic method, it was possible to annotate an average of 140 instances per hour. I annotated prepositions in Propbank sections two and three and section twenty three. My colleague Wang Pidong helped me to annotate section four. For section two and twenty three, all prepositional phrases were considered, whether they fill a semantic role in Propbank or not. For all other sections, I only considered prepositional phrases that span a semantic role in Propbank. To see how consistent humans can perform the annotation task, I re-annotated section 4 and computed the inter-annotator agreement between Pidong's and my annotation. The results showed that we agreed on the same sense in 86% of the cases, which is comparable to inter-annotator agreement of open-class words in the Penn Treebank in previous work [Palmer et al.2001]. Because I annotated running text, the prepositions are not equally represented, i.e. I did not annotate the same number of instances for every preposition. For the most frequent preposition `in`, I annotated 2094 instances, for the least frequent preposition `with`, only 483 instances. Table 4.1 shows the number of annotated instances for each preposition. Altogether, 6,681 instances for the seven most frequent prepositions were annotated. This makes the data set roughly half the size of the training and test material that was provided in the SemEval 2007 evaluation for the same prepositions. Although it was not possible to have all instances tagged by multiple annotators, I believe that the annotation is

```
<instance id="wsj_0201.s0.t13" docsrc="WSJ">
<context>
Rolls-Royce Motor Cars Inc.  said it expects its U.S. sales
to remain steady <head>at</head> about 1,200 cars in 1990 .
</context>
</instance>
```

Figure 4.1: Annotated WSD training instance from Propbank

reasonably accurate and consistent across different sections, as all annotation was either self-tagged or re-tagged by myself.

The manually annotated prepositions provide a gold standard for further experiments. They can either be used to evaluate the classifier that was trained on the SemEval training data or to re-train the model on domain specific preposition instances. I reserve section 23 for testing and use the other sections for training. This follows the same training/test data split that is common practice for SRL on Propbank. In total, there are 4767 instances for training and 1914 for test in the preposition gold standard. For every preposition, one XML file for training and one for test is created. The file format follows the format used in the SemEval data set. Every entry consists of an `instance` XML element that includes a unique identifier for the instance. The identifier consists of the file name in the Penn Treebank, the sentence number and the offset of the preposition in the sentence. The target prepositions is marked with a `<head></head>` XML tag in the context sentence. An example is shown in figure 4.1. The annotated sense labels for each preposition are stored in a separate file. The annotated instances from Propbank can be merged with the training data from SemEval to get a even larger training set. The experiments for the different training sets are presented in chapter 5.

Apart from using the gold annotation for training and testing the classifier, the annotation can help to analyze the correlation between semantic roles and preposition senses. A stronger correlation between roles and senses would make the sense more helpful for SRL. The bar diagram in figure 4.2 shows the distribution of coarse-grained senses of the most frequent preposition `in` among the semantic roles in the training sections. The most prominent correlation is between Argm-tmp and sense 2 ("expressing a period of time during which an event happens or a situation remains the case"). In 94.6% of the cases where `in` appears with a temporal adjunctive role, the sense of the preposition is sense 2. The correlation between Argm-loc and sense 1 ("expressing the situation of something that is or appears to be enclosed or surrounded by something else") and sense 5 ("expressing inclusion or involvement") is similarly strong. In 90% of the cases where the role Argm-loc appears with the preposition `in`, the preposition has either sense 1 or sense 5. The statistic confirms the initial motivation that spatial sense and locative arguments and temporal sense and temporal arguments would show a high level of agreement. It also revealed that most prepositions have a few dominating senses that appear far more frequent than the rest.

4.1.3 Predicting Propbank Function Tags

In an alternative setup, the same classifier can be used to predict the function tags of prepositional phrases in Propbank, e.g. loc, tmp, mnr, ext, etc. If one decides

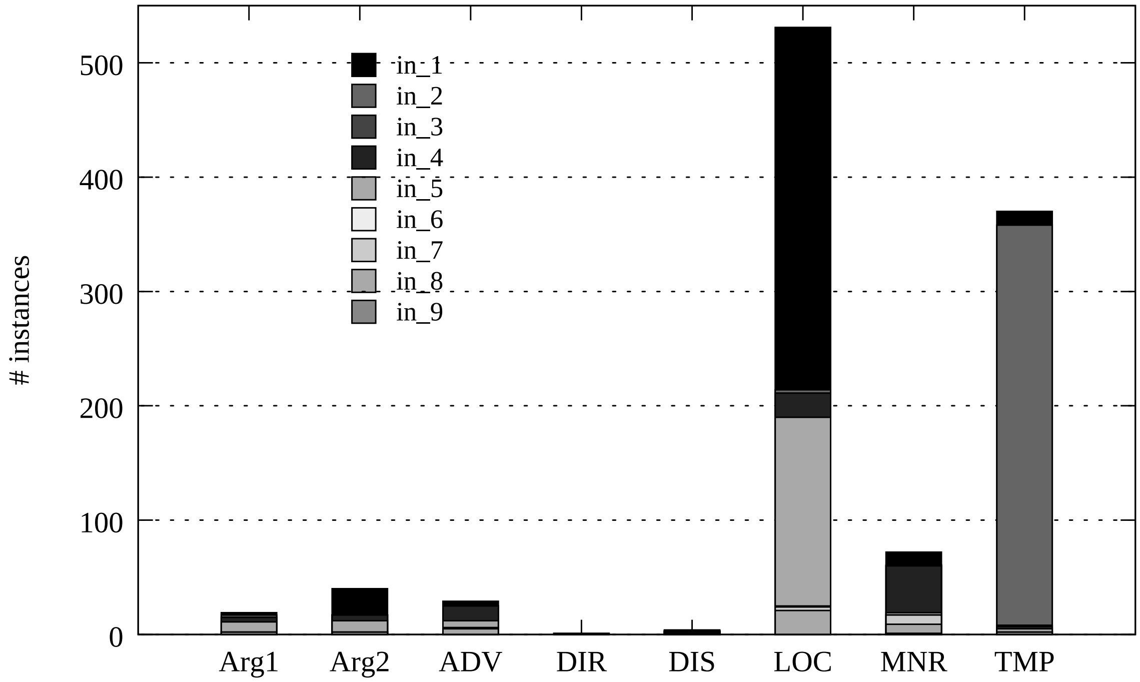


Figure 4.2: Distribution of coarse-grained senses for `in` among semantic roles in Propbank sections 2-4

Preposition	**Total**	**Training (sec 2-21)**	**Test (sec 23)**
at	1674	1549	98
for	1755	1658	97
in	7350	6907	443
of	414	387	27
on	1399	1308	91
to	323	305	18
with	864	813	51
Total	13,752	12,927	825

Table 4.2: Number of prepositional phrases with function tags in Propbank

to treat the function tag as a preposition sense, one gets sense annotated examples from Propbank "for free" and there is no need to manually tag prepositions. For each of the top seven prepositions, all prepositional phrases which have the preposition as the head word and are annotated as an adjunctive argument are automatically extracted. The sentence is kept as surrounding context and the position of the preposition in the sentence is marked. The function tag of the adjunctive argument label is taken as the sense of the preposition. This way, I get 12,927 instances for training from Propbank sections 2-21 and 825 instances for test from Propbank section 23. The detailed numbers of training and test instances are listed in table 4.2. Note that the test instances are much fewer than in the previous experiment, because in our manual annotation *all* prepositional phrases from section 23 were considered, not only those that are labeled as adjunctive arguments.

The WSD classifier can be used as a tagger to automatically label prepositions in Propbank with a (TPP or function tag) sense. However, the WSD classifier must not tag instances that it has previously seen during training. That would artificially inflate the accuracy of the classifier. Therefore, the tagging is performed in a cross-validation manner. The training instances are split into a number of equal bins. The classifier is trained on all except one bin and predicts labels for the remaining bins, thus avoiding training and tagging on the same instances. In my experiments, I used a split of 10 bins when tagging TPP preposition senses and 5 bins when tagging function tag senses.

The tagged preposition sense is used to investigate whether the sense is helpful in classifying the semantic role of prepositional phrases. The SRL experiments are described in the next chapter.

4.2 Semantic Role Labeling

This section describes an SRL system that uses preposition sense as an additional feature. The implementation was built on top of the SRL system that was developed by Prof Ng's student Liu Chang as part of his Honors Year Project.

The SRL system is a simpler version of the general SRL architecture that was presented in section 2.2.3. Instead of three, it only consists of two components: pre-processing and local scoring. The system does not attempt global scoring. The reason is that the relation between the preposition sense and the semantic role is local to the constituent. A prepositional phrases with the semantic role ARGM-LOC, for example, most likely occurs with a spatial sense for the preposition. But there is no linguistic constrain between the semantic role and other prepositions in the sentence. I believe that global scoring will not significantly benefit from the preposition sense information, so I concentrate my efforts on the local scoring step.

4.2.1 Pre-processing

In all experiments, I assume correct syntactic parse trees. The parse trees are taken from the Penn Treebank corpus. During pre-processing, the raw parse tree is extracted and merged with the semantic roles from Propbank into a single representation.

There are some special cases in the Propbank annotation that are not just labeled as ARG<NUMBER> or ARGM-<FUNCTION TAG>. The argument labels for these cases are standardized during pre-processing. The first special case consits of discontinuous arguments that span more than one node in the parse tree and co-referential nodes that point to other arguments. In my experiments, I treat discontinuous and co-referential arguments as separate instances of the same role. The C and R prefixes for discontinuous and co-referential arguments and are removed during pre-processing. The other special case consists of so-called null elements which do not have any surface realization on the word level. These nodes are not considered in this thesis either. Instead, empty nodes are removed during pre-processing. The Propbank labels are automatically re-aligned with the "cleaned" parse tree. The result of the pre-processing step is a annotated parse tree that has been stripped of superfluous prefix tags and null elements.

4.2.2 Local Scoring

The local scoring component is the core of the SRL pipeline where the SRL classifier predicts semantic roles for constituents. The SRL classifier in this work is based on maximum entropy models. Maximum entropy models have successfully been applied to a number of NLP tasks, including SRL. They achieve state-of-the-art accuracy and can be trained much faster than SVM models. Maximum entropy models do not make any independence assumptions about the features, which allows great flexibility in encoding linguistic knowledge via features. Features are encoded as binary-valued functions $f(x, y) \mapsto \{0, 1\}$ that map a tuple of input value $x \in X$ and output class $y \in Y$ to 0 or 1, indicating whether the feature is "active" for this class or not . The number of feature functions can be in the magnitude of hundreds of thousands. In this thesis, I use Zhang Le's Maximum Entropy Modeling Toolkit[1].

During training, the maximum entropy model learns a weight parameter for every feature, using maximum likelihood estimation (MLE). The learning algorithm is guaranteed to converge to the unique distribution that is both the maximum likelihood distribution and the maximum entropy distribution. That is, it satisfies the constraints of the training data, while being as uniform as possible otherwise. During testing, the maxent model computes the conditional probability $P(y|x)$ of the class y, given the observed features x.

In the context of SRL, the observed features x represent the syntactic parse tree t, the predicate p and the constituent node v that is under consideration. Thus, the maxent model tries to compute the conditional probability $P(l|t, p, v)$ of the label l, given the parse tree t, predicate p and constituent node v. For identification, all semantic roles except the NONE class are collapsed and the possible classes are ARG or NONE.

$$\begin{aligned} \hat{y} &= \operatorname*{argmax}_{y \in \{\textsc{Arg},\ \textsc{None}\}} P(y|t, p, v) \\ &= \operatorname*{argmax}_{y \in \{\textsc{Arg},\ \textsc{None}\}} P(y|\Phi(t, p, v)) \end{aligned} \tag{4.1}$$

Where $\Phi(\cdot, \cdot, \cdot)$ is a mapping from the parse tree, predicate and constituent to the feature space.

For classification, the model tries to compute the probability of a specific semantic role l, given the parse tree t, predicate p and constituent node v and the result of the identification step $\hat{y}$. The NONE class is excluded from the set of possible classes. The classifier outputs the semantic role l that receives the highest probability.

$$\begin{aligned} \hat{l} &= \operatorname*{argmax}_{l \in \mathcal{L} \setminus \{\textsc{None}\}} P(l|\hat{y}, t, p, v) \\ &= \operatorname*{argmax}_{l \in \mathcal{L} \setminus \{\textsc{None}\}} P(l|\hat{y}, \Phi(t, p, v)) \end{aligned} \tag{4.2}$$

I conduct experiments with two classifier models of different strength. The first model uses the seven baseline features that were first proposed in the original work of [Gildea and Jurafsky2002]. This model is referred to as the *weak baseline model*. If the preposition sense does not raise the performance above this baseline system, it would be extremely unlikely that the sense would result in a performance increase in other, more advanced SRL systems. The second model uses state-of-the-art features from other SRL systems. I use the same features as the system in [Jiang and Ng2006]. The features include the seven baseline features, additional

[1] http://homepage.inf.ed.ac.uk/s0450736/maxent_toolkit.html

Baseline Features [Gildea and Jurafsky2002]	
predicate	predicate lemma
path	syntactic path from constituent to predicate through the tree
phrase type	syntactic category (NP, PP, etc.) of the phrase
position	relative position to the predicate (right or left)
voice	whether the predicate is active or passive
head word	syntactic head of the phrase
sub-cat	the rule expanding the predicate node's parent
Advanced Features[Pradhan et al.2005]	
head POS	POS of the syntactic head
noun head PP	the head and phrase type of the rightmost NP child if the phrase is PP
first word	the first word and POS in the constituent
last word	the last word and POS in the constituent
parent constituent	the phrase type of the parent node
parent head	the syntactic head and POS of the parent node
right sister constituent	the phrase type of the right sister node
left sister constituent	the phrase type of the left sister node
right sister head	syntactic head and POS of the right sister
left sister head	syntactic head and POS of the left sister
temporal cue words	whether temporal key words are present
partial path	partial path from the constituent to the lowest common ancestor with the predicate
projected path	syntactic path with directions in the path removed
Feature Combinations [Xue and Palmer2004]	
predicate & phrase type	concatenation of predicate and phrase type
predicate & head word	concatenation of predicate and syntactic head
predicate & path	concatenation of predicate and path
predicate & position	concatenation of predicate and relative position

Table 4.3: Strong Baseline Features for SRL [Jiang and Ng2006]

features from [Pradhan et al.2005] and feature combinations that are inspired by the system in [Xue and Palmer2004]. In total there are 34 different features and feature combinations which are listed in table 4.3. This model is called the *strong baseline model.*

To reduce the number of features for training, some systems perform greedy feature selection. Starting from a baseline feature set, new features are incrementally added according to their contribution to the test accuracy on a development set. For this work, I decided to use all features for both the identification and the classification step without any additional feature selection. By using all the features, it can be ensured that the information of the preposition sense was not already encoded in another feature that was omitted due to feature selection.

To investigate the leverage of preposition sense for SRL, the preposition sense s is added as a feature to the weak and the strong baseline model. The identification and classification model then take the following form.

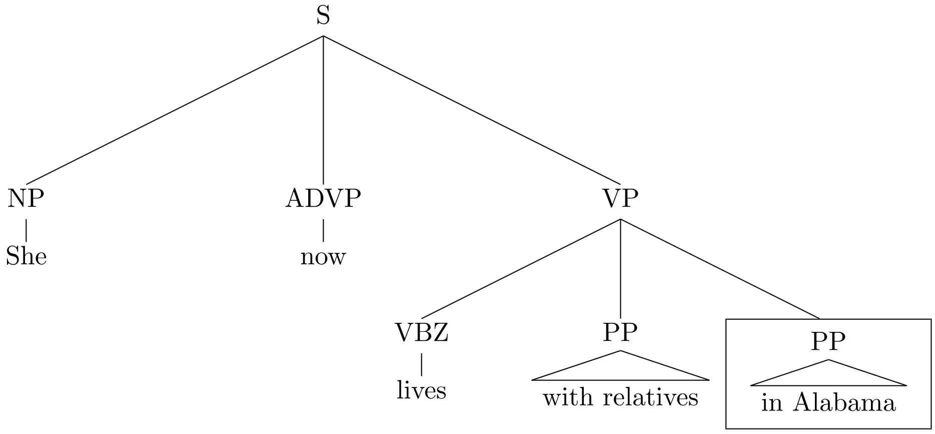
$$\hat{y} = \operatorname*{argmax}_{y \in \{\textsc{Arg},\ \textsc{None}\}} P(y|\Phi(t,p,v,s)) \tag{4.3}$$

$$\hat{l} = \operatorname*{argmax}_{l \in \mathcal{L} \setminus \{\textsc{None}\}} P(l|\hat{y}, \Phi(t,p,v,s)) \tag{4.4}$$

Depending on what kind of sense definition is used (TPP senses or function tags), there are three different features that can be incorporated in the maxent model:

Preposition Sense Features	
tpp_fine	Concatenation of lemma and fine grained TPP sense
tpp_coarse	Concatenation of lemma and coarse grained TPP sense
prop_label	Function tag sense

Table 4.4: Preposition Sense Features



(wsj_0281.s10)

Figure 4.3: Syntactic parse tree with prepositional phrase and preposition *in*

TPP fine-grained sense, TPP coarse-grained sense or function tag sense. The function tag sense is defined across prepositions and can directly be used as a feature. The fine and coarse grained TPP preposition sense are defined specifically for each preposition. The sense identifier is concatenated with the preposition lemma to form an unambiguous feature value, e.g. *in_1(1)* for the first fine grained sense of the preposition `in` or *in_1* for the first coarse grained sense. Table 4.4 lists the three different preposition features.

For illustration, consider the following example sentence from the Wall Street Journal with the predicate `lives` and the constituent `in Alabama`.

She now *lives* with relatives [in Alabama].

(wsj_0281.s10)

Figure 4.3 shows the syntactic parse tree for the sentence. Table 4.5 shows the instantiation of the strong baseline features and table 4.6 shows the instantiation of preposition sense features for the constituent `in Alabama`.

Although they use the same set of features, the models for identification and classification are trained differently. When training the identification model, every node in the parse tree provides one training example. All nodes that are labeled with a semantic role are positive training examples, the rest are negative examples. When training the classification model, correct identification is assumed. The model is only trained on constituents that are labeled with a not-none semantic role. Maximum entropy models can directly be trained for multi-class classification problems, so there is no need to break the problem into multiple binary classification problems. There are two parameters that have to be adjusted for maxent models: the number

Baseline Features [Gildea and Jurafsky2002]	
predicate	live
path	PP↑VP↓VBZ
phrase type	PP
position	right
voice	active
head word	in
sub-cat	VP>VBZ+PP+PP
Advanced Features[Pradhan et al.2005]	
head POS	VBZ
noun head PP	alabama NNP
first word	in IN
last word	alabama NN
parent constituent	VP
parent head	lives VBZ
right sister constituent	*nil*
left sister constituent	PP
right sister head	*nil nil*
left sister head	with IN
temporal cue words	no
partial path	PP↑VP
projected path	PP-VP-VBZ
Feature Combination [Xue and Palmer2004]	
predicate and phrase type	live_PP
predicate and head word	live_in
predicate and path	live↓PP↑VP↑VBZ
predicate and position	live_right

Table 4.5: Instantiation of strong baseline features for constituent `in Alabama`

Preposition Sense Features	
tpp_fine	in_1(1)
tpp_coarse	in_1
function tag sense	loc

Table 4.6: Instantiation of preposition sense features for constituent `in Alabama`

of training iterations and the Gaussian smoothing parameter. In most experiments, I kept the parameter settings from Liu Chang's implementation, unless stated otherwise. The number of training iterations was fixed to 500 and the Gaussian smoothing parameter was set to 0, which means that no smoothing is performed.

The effect of the preposition sense is investigated by adding the new feature to the (weak or strong) baseline model, re-training the models and measuring the change in performance on the test set. If there is a relation between preposition sense and semantic roles, there should be a change in the performance, unless the information is already encoded somewhere in the other features. The results for the experiments can be found in the next chapter.

Adding the preposition sense as a feature to the classifier is the most straightforward way to integrate this knowledge source into the SRL model, but not the only possible way. I investigate two alternative methods to integrate the sense in the SRL pipeline, one is inspired by classifier combination, the other by joint learning.

4.3 Classifier Combination for SRL

As we have seen in section 2.3.2, function tags of adjunctive arguments in Propbank can be treated as a coarse set of preposition senses. The function tag preposition sense can be predicted by a WSD classifier.

It can be observed that for prepositional phrases that fill adjunctive roles, the WSD classifier and the SRL classifier (by definition of the preposition sense) try to predict the same class. The two classifiers approach the problem from two different perspectives: The SRL classifier tries to find the semantic role of the phrase with respect to a predicate; the WSD classifier tries to predict the sense of the head preposition.

We could use *both classifiers* to predict the semantic role of the adjunctive constituents. Instead of using the predicted sense as a feature, both models could be combined trough *classifier-combination*. The motivation behind classifier combination is that different classifiers have different strengths and will perform differently on different test instances. The difference in the classifiers derives from different machine learning algorithms and different features. The SRL classifier uses a broad set of syntactic features from the parse tree that encode information about the predicate and the current constituent. The WSD classifier uses features from the surrounding words and POS. The classifiers also use different learning algorithms (maxent models and SVM). It is reasonable to assume that the prediction errors of the classifiers are not too closely correlated.

If the classifier-combination model succeeds in combining the strengths of the individual systems without ruling out correct answers, the combined model will show a better performance than the individual models. Classifier combination has been applied to WSD to combine the output of multiple classifiers [Brill and Wu1998, Stevenson and Wilks2001].

When we combine SRL and WSD, the situation is slightly more complicated, because the combination model is only applicable for adjunctive semantic roles. That is why the system first needs to identify those arguments that ought to be considered in the combination model. The SRL pipeline has to be extended by another binary classification step that predicts whether an argument is an adjunctive prepositional phrase or not. This step is referred to as *adjunctive-PP* classification. Deciding whether an argument is a prepositional phrase or not is determined by the syntactic phrase type of the constituent; deciding whether the argument is an adjunctive argument or a core argument, on the other hand, is not that straightforwards. Many prepositional phrases appear as core roles of the predicate. The verb `settle`, for example, can take an extra core argument to denote the end point of the settling. The level that something comes to rest at is labeled as ARG4 and not, for example, as ARGM-LOC.

> [The March delivery, which has no limits,]$_{\text{ARG1}}$ [settled]$_{rel}$
> [at 14.53 cents , up 0.56 cent a pound]$_{\text{ARG4}}$.
>
> (wsj_0155.s4)

I train a maxent classifier with the same features as the SRL classification model for the adjunctive-PP classification problem. Again, correct parse trees and correct argument boundaries are assumed. All adjunctive arguments that are prepositional phrases contribute the positive examples, all other arguments are negative examples. During testing, all instances that are classified as adjunctive prepositional phrases are passed to the combination model.

The combination model can be learned by a classifier. The input to the classifier are the predictions of the SRL and WSD classifier, plus the predicate lemma and the preposition. The intuition here is that the two classifiers will perform differently for different verbs and prepositions. For example, the WSD classifier might be very accurate for the preposition `at` and should be given more confidence in cases where `at` appears in the phrase. On the other hand, the SRL classifier might be very accurate for certain verbs. By including the verb lemma and the preposition as a feature for the combination model, the combination classifier is given the chance to learn such rules.

As the learning algorithm for the combination model, I chose a decision tree classifier[2]. A decision tree classifier is a hierarchical model that breaks the classification task into a number of simple decision rules. I also conduced experiments with a number of other combination methods like bagging or Bayesian networks. I found that their performance was slightly lower than the decision tree classifier.

The combination model is trained on the predictions of the SRL classifier and the WSD classifier on all adjunctive prepositional phrases in the training set. The generation of training data is performed in a cross-validation manner by splitting Propbank sections 2-21 into five equal bins and training and testing the SRL and WSD model on each fold in turn. During testing, the SRL pipeline includes the following five steps:

1. **Identification:** Identify all arguments of the predicate
2. **Adjunctive-PP classification:** Split arguments into adjunctive PP and others
3. **Classification:** Assign semantic roles for all arguments

[2] I used the decision tree implementation in WEKA

4. **Preposition WSD:** Predict function tags for all PP that were identified as adjunctive arguments

5. **Combination:** Combine both predictions in the combination model

For all prepositional phrases which are not classified as adjunctive arguments the semantic role that is assigned during the classification step is the final prediction.

4.4 Joint Learning of Semantic Roles and Preposition Senses

In the two models that are presented above, SRL and preposition WSD are processed in a pipeline of sequential classification steps.

The pipeline approach has a number of drawbacks. Only the 1-best predicted preposition sense is used in the following SRL step. Errors in the preposition disambiguation step are carried on to the SRL step and introduce noise to the system. If the accuracy of the preposition sense classifier is low, the noise might have a negative effect on the following SRL step.

Instead of learning the WSD and the SRL problem separately, we would like to learn a model that maximizes the *joint probability* of the semantic role and the preposition sense. The semantic role l and preposition sense s that maximize the probability, given the parse tree t, predicate p, constituent node v and surrounding context c, can be written as follows.

$$\widehat{(l,s)} = \underset{\text{(l,s)}}{\operatorname{argmax}}\ P(l,s|t,p,v,c) \tag{4.5}$$

A simple application of the chain rule allows us to factor the joint probability into a SRL and a WSD component.

$$\widehat{(l,s)} = \underset{\text{(l,s)}}{\operatorname{argmax}}\ P(l|t,p,v,c) \times P(s|t,p,v,c,l) \tag{4.6}$$

I assume that the probability of the semantic role is already determined by the syntactic parse tree, the predicate and the constituent node and conditionally independent of the remaining surrounding context. Likewise, I assume that the probability of the preposition sense is conditionally independent of the parse tree, predicate and constituent, given the surrounding context and the semantic role of the dominating prepositional phrase.

$$\begin{aligned}
\widehat{(l,s)} &= \underset{\text{(l,s)}}{\operatorname{argmax}}\ P(l|t,p,v,c) \times P(s|t,p,v,c,l) \\
&= \underset{\text{(l,s)}}{\operatorname{argmax}}\ P(l|t,p,v) \times P(s|c,l) \\
&= \underset{\text{(l,s)}}{\operatorname{argmax}}\ P(l|\Phi(t,p,v)) \times P(s|\Psi(c,l)) \qquad (4.7)
\end{aligned}$$

where $\Phi(\cdot,\cdot,\cdot)$ and $\Psi(\cdot,\cdot)$ are task specific feature maps for SRL and WSD respectively.

We observe that the first component in the joint model corresponds to equation 4.2 in the maxent model for SRL, if we assume that the constituent is identified as an argument. The second component seeks to maximize the probability of the preposition sense, given the context and the semantic role of the dominating phrase. Both components can be chained together by simply multiplying the probabilities, but we need models that output a full probability distribution during classification. The preposition sense classifier that we presented earlier in this chapter is based on SVM that unfortunately do not output a probability distribution. So I re-implemented the preposition WSD classifier based on maxent models. The maxent WSD model uses the same features as the previous classifier plus the semantic role of the dominating prepositional phrase. During training, the Propbank gold SRL label is used. During test, the system relies on the maxent SRL model to automatically predict the semantic role. The joint model is trained on Propbank sections 2-4 which are jointly annotated with semantic roles and preposition senses. During testing, the classifier seeks to find the tuple of semantic role and preposition sense that maximizes the joint probability. The classifier computes the probability of each semantic role, given the SRL features and multiplies it with the probability of the most likely preposition sense, given the semantic role and the WSD features. The tuple that receives the highest joint probability is the final output of the joint classifier.

I compare the joint model against two other models for each task: and independent baseline model and an extended basline model. The independent baseline model only uses task specific features. For SRL, the baseline model is the strong baseline SRL model from section 4.2. For WSD, the baseline model is the maxent WSD classifier with the three knowledge sources POS, bag of words and local collocations. The extended baseline model uses the most likely prediction of the other task as an additional feature. For SRL, the extended baseline model is the strong baseline model which receives the coarse-grained preposition sense as an additional feature. For WSD, the extended baseline model is the maxent WSD model with the same features as before, plus the semantic role of the constituent. During training the additional feature is taken from the joint annotated gold labels, during testing the additional feature is automatically predicted. All models are trained on sections 2 to 4 to make results comparable. The models are tested on all prepositional phrases in Propbank section 23. The number of training iterations and the Gaussian smoothing parameter for the maxent models are tuned through 10-fold cross-validation on the training set for every model

4.5 Summary

This chapter gave a detailed description of the methods that were applied to preposition WSD and SRL as part of the research work in this thesis. The main contribution is the empirical evaluation of different models that use preposition sense to determine the semantic role of a constituent. I presented three different approaches: adding preposition sense as a feature, classifier combination and joint learning. Because the preposition sense is not readily available in the Propbank corpus, it had to be annotated first. The chapter described the WSD classifier that was used to tag prepositions in Propbank and the manual annotation process of 6,681 preposition instances. The work in this thesis should help to clarify whether the preposition sense is useful for SRL or not. The experiments and results are presented in the next chapter.

5. Experiments and Results

The annotation of preposition senses in Propbank confirmed the hypothesis that there is a strong correlation between semantic roles and preposition senses. However, it is not clear whether the tasks can actually benefit from each other. In this chapter, I present a number of different experiments to test the hypothesis that preposition sense can help SRL.

5.1 Evaluation Metrics

Even the best experiments become useless if one does not know how to interpret the results. This section describes the evaluation metrics that are reported in the experiments. The three key metrics that are used to report the results in many NLP tasks are *accuracy*, *precision*, *recall* and *F1 measure*. Which metric is applicable depends on the type of the experiment. In NLP classification problems, one often faces the situation that only a small subset of the test instances is relevant to the target problem. In SRL, for example, one is only interested in those constituents that fill not-NONE semantic roles. In such a case, reporting the percentage of all correct predictions, would result in artificially high results with very limited predictive power, because the NONE arguments greatly outnumber all other semantic roles.

Instead, one might be interested in the percentage of *relevant* instances that were successfully classified. This metric is called *recall (r)* and is computed as the number of correctly retrieved relevant instances, divided by the number of all relevant instances in the corpus.

$$r = \frac{|\{\text{relevant instances}\} \cap \{\text{retrieved instances}\}|}{|\{\text{relevant instances}\}|} \tag{5.1}$$

This metric alone does not have much predictive power either, because recall does not measure how accurate classifier retrieves instances. That is why recall is always reported together with *precision (p)* which is the percentage of *retrieved* instances that were correctly classified. It is computed as the number of correctly retrieved relevant instances over the total number of retrieved instances.

$$p = \frac{|\{\text{relevant instances}\} \cap \{\text{retrieved instances}\}|}{|\{\text{retrieved instances}\}|} \tag{5.2}$$

The two values can be combined into a single metric called *F1 measure (F1)* which is computed as the harmonic mean of precision and recall.

$$F1 = 2 * \frac{p * r}{p + r} \tag{5.3}$$

F1 measure is also called balanced f measure, because precision and recall are weighted equally.

If all instances are relevant, there is no difference between precision and recall. In this case, the performance of the classifier is simply measured by the number of correctly classified instances divided by the total number of instances. This metric is called *accuracy.*

$$a = \frac{|\{\text{correct instances}\}|}{|\{\text{all instances}\}|} \tag{5.4}$$

In this thesis, accuracy is used to report the results for WSD and SRL classification. Precision, recall and F1 measure are used to report the performance of the models for SRL identification and classification of individual semantic roles. A SRL prediction is considered correct if the correct semantic role is assigned and the argument boundaries align with the boundaries of the gold standard. Otherwise, the prediction is considered as incorrect. The following section reports the results of the experiments for preposition sense disambiguation.

5.2 Preposition Sense Disambiguation

In many of the experiments, the SRL system relies on an automatic tagger to annotate the sense label for each preposition. Errors in the preposition sense disambiguation step will necessarily increase the noise in the system. The quality of the tagging is of great importance for the following SRL step. That is why my first experiments only test how accurate the preposition WSD classifier performs the tagging task.

5.2.1 TPP Preposition Sense

The TPP sense definitions were used in the SemEval 2007 word sense disambiguation task for prepositions which provided a large number of annotated training instances for each preposition. I refer to the SVM model which is trained on the SemEval training set as *WSD model 1.* Alternatively, I can use the instances that were manually annotated in three Propbank sections to train the model. I call this model *WSD model 2.* This model does not suffer from the cross-domain problem, but has fewer training examples to learn from. Finally, the manually annotated training instances from Propbank can be combined with the SemEval training data. The model learned from the combined data set is called *WSD model 3.*

I conduct experiments with each of the three models to investigate which results in the most accurate preposition sense tagger. In all experiments, the classifier was tested on the manually annotated preposition instances from Propbank section 23. The detailed scores for fine-grained and coarse-grained senses are given in table 5.1, together with the baseline accuracy when every preposition is assigned its most frequent sense (sense-one baseline). Figure 5.1 shows the coarse-grained accuracy of the different WSD models for the seven prepositions on the test set in section 23.

Preposition	**Test Instances**	**Sense-one**		**WSD model 1**		**WSD model 2**		**WSD model 3**	
		fine	coarse	fine	coarse	fine	coarse	fine	coarse
at	169	0.296	0.296	0.556	0.556	0.746	0.746	0.811	0.811
for	220	0.277	0.277	0.345	0.382	0.436	0.445	0.459.	0.473
in	595	0.311	0.318	0.371	0.402	0.650	0.667	0.659	0.682
of	320	0.309	0.341	0.331	0.362	0.581	0.609	0.653	0.619
on	186	0.226	0.226	0.247	0.446	0.414	0.548	0.430	0.618
to	269	0.257	0.260	0.245	0.335	0.550	0.599	0.517	0.599
with	155	0.284	0.284	0.310	0.335	0.471	0.497	0.490	0.535
Total	1914	0.287	0.295	0.343	0.396	0.571	0.604	0.587	0.635

Table 5.1: Accuracy of WSD models on prepositions in Propbank section 23

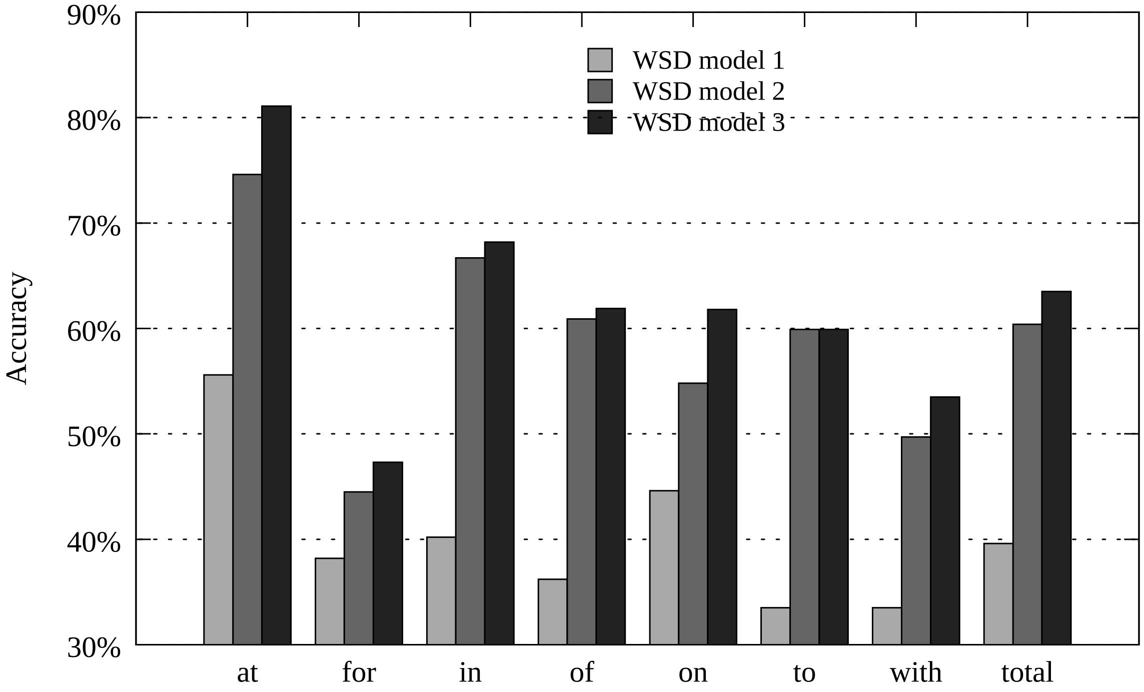


Figure 5.1: Coarse-grained classification accuracy of the WSD models on prepositions in Propbank section 23

Preposition	**Test Instances**	**Sense-one**		**MELB-YB**	**WSD model 1**	
		fine	coarse	fine	fine	coarse
at	367	0.425	0.425	0.790	0.787	0.804
for	478	0.238	0.238	0.573	0.672	0.684
in	688	0.362	0.493	0.561	0.612	0.683
of	1478	0.205	0.315	0.681	0.709	0.771
on	441	0.206	0.281	0.624	0.594	0.746
to	572	0.322	0.327	0.745	0.705	0.717
with	578	0.247	0.394	0.699	0.697	0.772
Total	4602	0.269	0.350	0.666	0.684	0.743

Table 5.2: Accuracy of WSD model 1 and the MELB-YB system on the official SemEval test set

To make my WSD classifier comparable to previous systems, I also report the results of WSD model 1 on the official SemEval test set in table 5.2 together with the official fine-grained results for the MELB-YB system [Ye and Baldwin2007], which was the best performing system in the competition. In the SemEval 2007 competition, only the fine-grained accuracy was reported for each preposition, so we cannot compare the coarse-grained accuracy. The results show that the classifier severely suffers from cross-domain adaptation when trained on the SemEval training set and tested on Propbank. The WSD model 1 achieves a fine-grained accuracy of 68.4% and coarse-grained accuracy of 74.3% on the SemEval test set, which is 2% better than the MELB-YB system. But when the model is tested on the Propbank instances, it only achieves a fine-grained accuracy of 34.3% and coarse-grained accuracy of 39.6%. That is only about 6% and 10% percent points above the sense-one baseline and only about half as accurate as the results the model achieves on the SemEval test set. This suggests two possible explanations: that the distribution of preposition senses differs severely across domains, and that the SemEval training data does not reflect the distribution of preposition senses in running text very well. Either way, an accuracy of 30-40% is not satisfactory and is most likely not high enough to improve the performance for SRL.

The WSD model 2 model avoids the domain adaptation problem, because the model is trained on instances from the same corpus and from running text. However, the accuracy is about 10%-15% lower compared to the SemEval results and the accuracy for different prepositions varies greatly. The difference between the score for the most and least accurate preposition is over 30%. That is about 10% larger than the span between the results for the same prepositions on the SemEval task. The reason is the unequal number of training instances in the preposition gold standard. The preposition `in`, which has the most training instances, was classified correctly in 65% of the cases (fine-grained), which is similar to the results in the SemEval task. This suggests that prepositions in Propbank can be disambiguated with the same accuracy as in the SemEval competition, provided that the same amount of training data is available. For those prepositions that have less training data, the results are generally lower. The prepositions `with` and `on`, which have less than 400 training instances, achieve fine-grained scores of 47.1% and 41.4% only. This is about 20% lower than the accuracy these prepositions achieve on the SemEval test set.

Sparseness of training data is a general problem in WSD. When there are many ambiguous words and each word has many senses, it is difficult to create a large number of training examples for each sense. The third experiment investigates if the

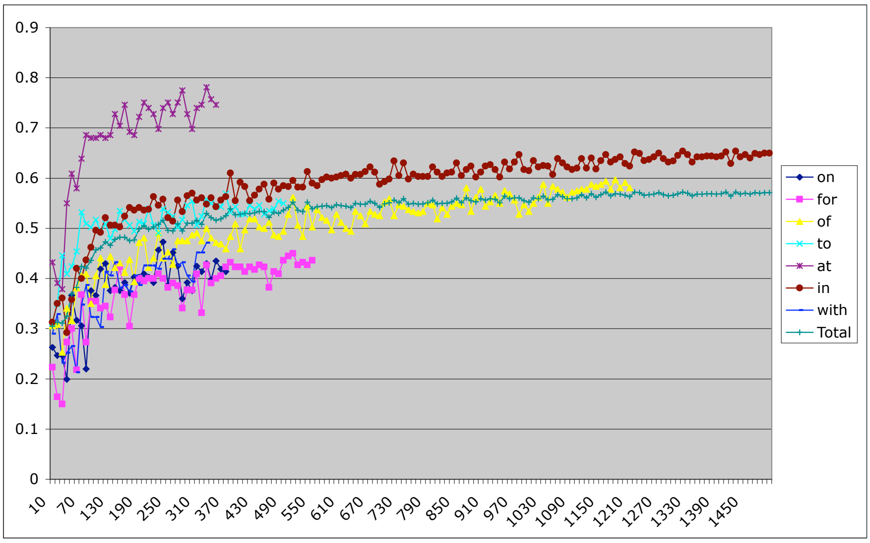


Figure 5.2: Learning curve for WSD model 2 on prepositions in Propbank section 23, fine-grained sense

data sparseness problem can be reduced by adding instances from the SemEval task to the training data. The total fine-grained accuracy of WSD model 3 is 1.6% higher than the accuracy of WSD model 2. The accuracy for the prepositions `with` and `on` increased by 1.9% and 1.6% respectively. As the WSD model 3 showed the best performance, it was used as the preferred model to automatically predict preposition senses in the following SRL experiments.

The large amount of additional training data on WSD model 3 only resulted in a modest increase in accuracy, because the additional data is not domain specific. The question is, whether we could hope for a larger improvement if more domain specific training instances from Propbank were added and hence whether future annotation would be meaningful or not.

To investigate the effect of the size of training data on the classification accuracy, WSD model 2 was re-trained on the gold annotated Propbank instances, but this time, the number of training instances was gradually increased to get a learning curve. The training sets are drawn randomly from the set of training instances for each run. The model is tested on all instances from section 23. The learning curves for fine- and coarse-grained sense are shown in figure 5.2 and 5.3 respectively .

The experiment shows that the classifier improves rapidly in the beginning, but already after about 250 training instances the improvement steps get significantly smaller. The irregular oscillation of the curve is a result of the randomly selected training data. The shape of the learning curve roughly resembles a logarithmic curve: doubling the training data results in a constant improvement in accuracy of roughly 3-5%. Hence, the amount of training data that has to be added to significantly improve the classifier grows exponentially when the training set gets

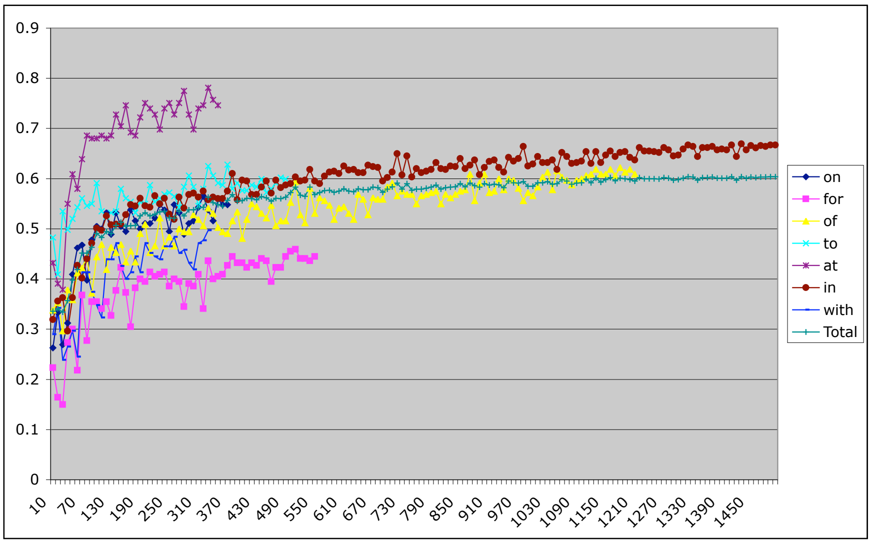


Figure 5.3: Learning curve for WSD model 2 on prepositions in Propbank section 23, coarse-grained sense

larger. The experiment shows that an automatic preposition sense tagger needs a large annotated training corpus from the same domain to achieve accurate results.

The WSD model 3 achieves an accuracy that is comparable with the results on the SemEval WSD task, if enough training data is available. This shows that prepositions in Propbank can be disambiguated with state-of-the-art accuracy. However the experiments also show that the effect of additional training data is limited. Further annotation of training data should therefore first concentrate on those prepositions which have fewer training instances.

5.2.2 Propbank Function Tags

This sub-section describes the WSD experiments with the alternative preposition sense definition that is induced by the function tags in Propbank. All prepositional phrases from the Propbank corpus that fill the role of an adjunctive argument and have one of the top seven prepositions as the lexical headword were automatically extracted. The function tag of the semantic role is treated as the sense of the preposition. Sections 2 to 21 are used for training and section 23 is kept for testing. Because training and test data are all taken from the same corpus, domain adaptation is not a problem in this experiment.

The results for disambiguating the function tag are shown in table 5.3. The average accuracy of the SVM classifier on the function tag disambiguation task is 84.5%. There is no differentiation between fine- and coarse-grained senses, as the “sense” is just the atomic function tag label.

This result is significantly higher than the results in the previous experiments with TPP preposition senses. This is not surprising if we consider that the function tag

Preposition	**Test Instances**	**Sense-one**	**Function Tag Model**
at	98	0.306	0.898
for	97	0.351	0.763
in	443	0.470	0.874
of	27	0.556	0.926
on	91	0.462	0.714
to	18	0.444	0.889
with	51	0.804	0.824
Total	825	0.458	0.845

Table 5.3: Accuracy of the WSD classifier on the function tag disambiguation task on prepositions in Propbank section 23

sense is much coarser than the TPP preposition sense and that there is more training data available to learn the model. The average number of possible function tags for the seven prepositions is nine and there are only twelve function tags in total. In contrast, the average number of possible (fine-grained) preposition senses is 13.6 and some prepositions can have more than twenty senses.

The training data for the function tag classifier is about 2.7 times the amount of training data that was annotated with TPP senses for WSD model 2.

Another more subtle reason why the accuracy in this experiment is possibly higher, is that there might be a selection bias in the test data. It is possible that those prepositional phrases that fill semantic roles in Propbank are more “typical” uses of the preposition and easier to disambiguate than other prepositional phrases. Still, the experiment shows that the classifier can disambiguate the function tag of a prepositional phrase in Propbank with a accuracy of over 80%. My results are comparable to those that [O’Hara and Wiebe2003] and [Ye and Baldwin2006] reported in similar experiments.

The question in the following experiments is whether the automatic annotation is able to boost the performance of a SRL classifier.

5.3 Semantic Role Labeling

In these experiments, I investigate the leverage of preposition sense as a feature in a maxent SRL classifier. The experiments are conducted with two models of different strength: a *weak baseline* and a *strong baseline* model. The difference is the feature set they use. The weak model is trained with the seven basic features which were proposed by [Gildea and Jurafsky2002]. The strong baseline is trained with a much larger set of 34 features which are taken from [Jiang and Ng2006]. The features in this model are basically state-of-the-art.

5.3.1 Weak Baseline Model

The first set of experiments investigates the effect of the preposition sense features on the performance of the weak baseline model. The maxent models for argument identification and classification are trained on sections 2 to 21 from the Propbank corpus and tested on section 23. Depending on the type of preposition sense and the level of granularity, there are three possible preposition sense features:

Model	**Classification**	**Identification**		
	A	P	R	F1
Weak Baseline	83.91	93.84	94.14	93.99
+ fine	86.80	94.28	94.62	94.45
+ coarse	86.89	94.06	94.59	94.32
+ function tag	87.50	94.29	93.74	94.01

Table 5.4: Effect of the preposition sense feature on the weak baseline model for SRL on Propbank section 23, trained on Propbank sections 2-21

- fine-grained TPP preposition sense,
- coarse-grained TPP preposition sense and
- function tag preposition sense

The preposition sense feature is automatically predicted by the SVM classifier with WSD model 3 that was described in the previous section. Because WSD model 3 is only trained for the prepositions `at, for, in, of, on, to` and `with`, only the sense for these seven prepositions can be included. They cover about 38% of all prepositional phrases in Propbank section 2 to 21. In other experiments with WSD model 1, I found that the difference in SRL performance was marginal when using all 34 prepositions and using only the seven most frequent prepositions.

The results of the experiment are listed in table 5.4. The accuracy of the weak baseline model for argument classification is 83.91%. Adding the new preposition feature improves the classification accuracy by about 3%. For identification, the sense features improves the F1 measure only marginally ($< 0.5\%$).

The results match our expectation. We have previously seen that there is a strong correlation between the preposition sense and the semantic role of the dominating constituent. Therefore, the preposition sense should have a positive effect on the classification accuracy. For argument identification we do not necessarily expect a significant effect. The reason is that the sense of the prepositions is discriminative between different semantic roles, but not between arguments and non-arguments. Consider the prepositional phrase `in the dark` in the sentence:

> “We’re in the dark,” he said.
>
> (wsj_1803.s45)

The phrase is clearly not an argument to the predicate `say`. But if we alter the syntactic structure of the sentence appropriately, the same phrase suddenly becomes an adjunctive argument: In the dark, he said “We are”. On the other hand, we can easily find other prepositional phrases where `in` has a different meaning, but the phrase always fills a semantic role: In a separate manner, he said . . . , In 1998 he said . . . , In Washington, he said . . . , etc. This illustrates that the preposition sense is not determined by whether the constituent is an argument or not.

Although the SRL classification accuracy improved in this experiment, the impact of the sense feature is limited, because only a minority of the arguments in Propbank are prepositional phrases. I found that about 15% of the arguments in Propbank section

Semantic Role	Test Instances	Baseline	+Fine	+Coarse	+Function Tag
Overall	14508	83.91(A)	86.80(A)	86.89(A)	87.50(A)
ARG0	3810	93.72	94.55	94.50	94.72
ARG1	5467	89.42	91.18	91.30	91.50
ARG2	1140	63.74	69.58	70.19	69.66
ARG3	177	52.28	58.75	56.73	55.32
ARG4	103	82.86	80.00	78.85	80.89
ARG5	5	50.00	88.89	88.89	88.89
ARGM-ADV	509	61.24	70.93	70.76	70.85
ARGM-CAU	77	64.75	66.20	63.38	69.06
ARGM-DIR	86	50.63	63.35	62.58	62.65
ARGM-DIS	319	77.80	81.34	81.53	85.44
ARGM-EXT	35	52.46	61.02	62.07	66.67
ARGM-LOC	384	54.64	66.06	65.71	72.68
ARGM-MNR	352	51.01	61.23	60.80	61.80
ARGM-MOD	554	99.46	99.28	99.46	99.55
ARGM-NEG	229	99.34	98.70	99.35	99.35
ARGM-PNC	115	47.44	58.99	59.43	57.41
ARGM-PRD	5	0.00	28.57	28.57	28.57
ARGM-REC	2	0.00	0.00	0.00	0.00
ARGM-TMP	1139	76.26	82.47	83.19	85.27

Table 5.5: Detailed scores for the weak baseline model and preposition sense feature on the SRL classification task on Propbank section 23, trained on Propbank sections 2-21

2 to 21 are prepositional phrases. Especially for the core arguments Arg0 and Arg1, which are the most frequent in Propbank, there are only a few prepositional phrases. For these arguments, it is obvious that the new feature will not have an effect. For adjunctive arguments that often appear as prepositional phrases such as Argm-loc, Argm-ext or Argm-tmp, I expect the feature to be more helpful.

To get a better idea of the utility of the preposition sense feature for individual semantic roles, I computed the F1 measure for each semantic role. Table 5.5 shows the detailed scores of the experiment. As expected, the improvement for adjunctive arguments is greater than for core arguments. Figure 5.4 displays the F1 measure for the semantic roles Argm-loc and Argm-tmp and the overall accuracy. The improvement for the semantic role Argm-loc is about 18% and the improvement for Argm-tmp is about 9%. It can be observed that the coarser function tag sense yields slightly better results than the TPP sense. It is satisfying to see that the preposition sense proves to be a strong feature for adjunctive argument that describe the time or location of the action, as this was part of the initial motivation for this work.

5.3.2 Strong Baseline Model

Although the weak baseline model showed that preposition sense is a useful feature for SRL classification, the results would only have practical impact if the preposition sense helped to improve the performance of a state-of-the-art SRL system. In the second set of experiments, I investigate if the preposition sense feature still shows a positive effect when added to a state-of-the-art SRL system. The baseline classification accuracy for the strong baseline model is 91.78%, significantly higher than the weak baseline. The experiments are conducted analogously to the experiments on the weak model. The only difference is that the strong baseline model uses the

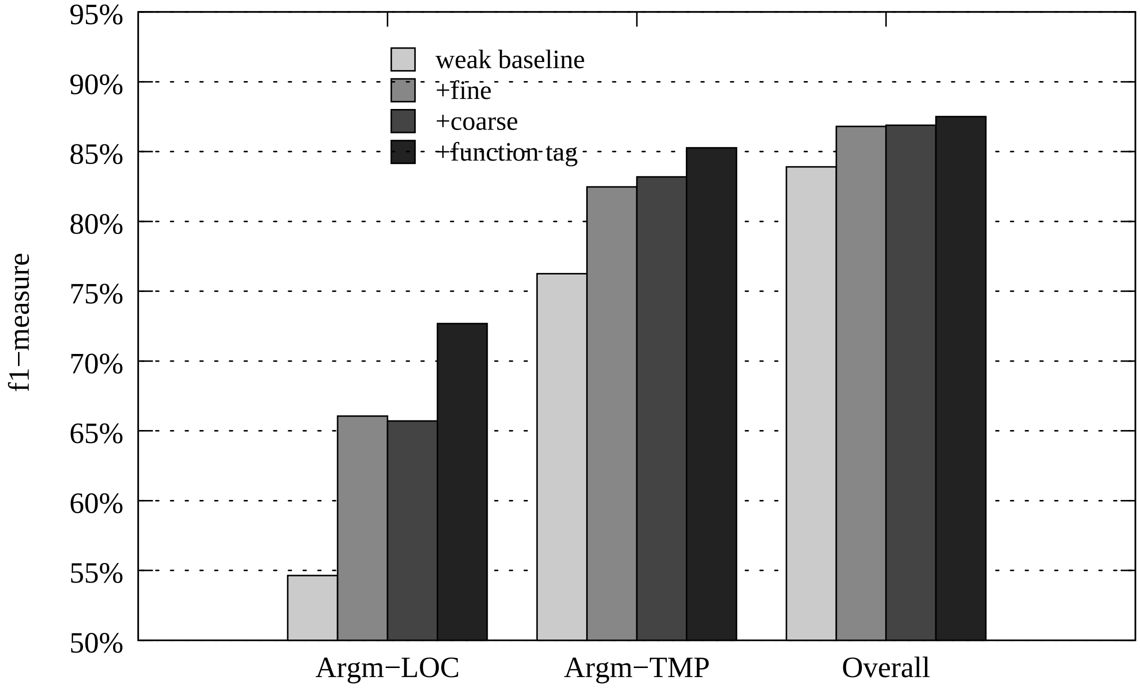


Figure 5.4: F1 measure of the weak baseline model for Argm-loc, Argm-tmp and overall accuracy on the SRL classification task on Propbank section 23

Model	Classification	Identification		
	A	P	R	F1
Strong Baseline	91.78	94.63	96.03	95.32
+ fine	91.54	94.70	95.91	95.30
+ coarse	91.35	94.86	96.04	95.44
+ function tag	91.78	94.82	96.03	95.42

Table 5.6: Effect of the preposition sense feature on the strong baseline model for SRL on Propbank section 23, trained on Propbank sections 2-21

more sophisticated feature set of 34 features (see table 4.3). The results are shown in table 5.6. Adding the preposition sense feature does not lead to an improvement, instead it causes the accuracy to drop slightly (TPP sense) or remain unchanged (function tag sense). Figure 5.5 shows the effect for the semantic roles Argm-loc, Argm-tmp and the overall accuracy. All three sense features fail to improve the performance.

The detailed scores are shown in table 5.7. The scores show that the TPP sense feature results in a drop of performance for almost all semantic roles. For the function tag sense, the performance for some roles increases compared to the baseline, yet the feature fails to improve the overall classification accuracy. The effect can be explained with the noise that is introduced by the automatic sense tagger. Because the accuracy of the tagger is only around 60%, the noise level is very high. For the function tag sense, the accuracy of the tagger is over 80%, but still stays behind the accuracy of the SRL classifier.

The question is whether the preposition sense could still improve the classification accuracy if we had a more accurate preposition sense tagger. In that case, we could

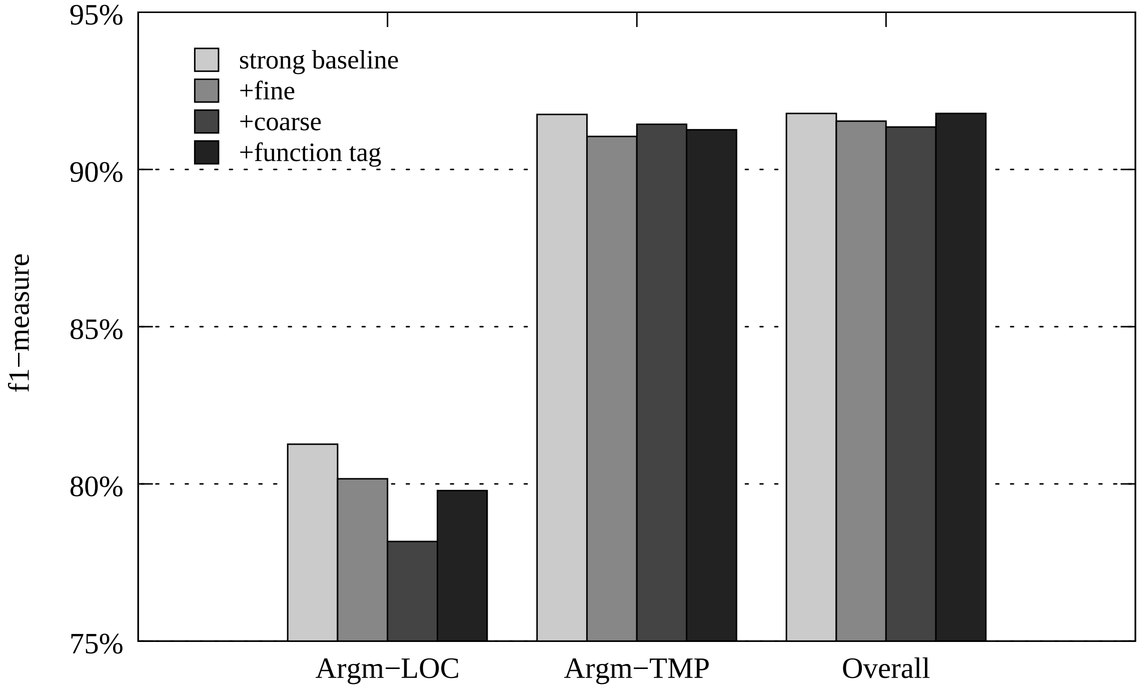


Figure 5.5: F1 measure of the strong baseline model for Argm-loc, Argm-tmp and overall accuracy on the SRL classification task on Propbank section 23

Semantic Role	Test Instances	Baseline	+Fine	+Coarse	+Function Tag
Overall	14508	91.78(A)	91.54(A)	91.35(A)	91.78(A)
ARG0	3810	95.74	95.73	95.34	95.76
ARG1	5467	94.83	94.79	94.69	94.94
ARG2	1140	85.10	84.95	84.62	85.56
ARG3	177	78.08	78.05	77.68	77.91
ARG4	103	84.76	85.99	84.21	86.41
ARG5	5	90.91	90.91	100.00	100.00
ARGM-ADV	509	74.90	73.61	72.33	74.79
ARGM-CAU	77	78.32	77.46	78.08	80.54
ARGM-DIR	86	78.31	76.07	77.11	80.92
ARGM-DIS	319	89.16	88.96	88.24	89.66
ARGM-EXT	35	64.41	63.33	64.41	60.00
ARGM-LOC	384	81.26	80.16	78.17	79.79
ARGM-MNR	352	69.05	67.30	68.98	68.05
ARGM-MOD	554	99.64	99.55	99.64	99.55
ARGM-NEG	229	99.35	99.35	99.35	99.35
ARGM-PNC	115	67.58	64.29	64.89	62.78
ARGM-PRD	5	28.57	25.00	28.57	33.33
ARGM-REC	2	0.00	0.00	0.00	0.00
ARGM-TMP	1139	91.75	91.05	91.44	91.26

Table 5.7: Detailed scores for the strong baseline model and sense feature on the SRL classification task on Propbank section 23, trained on Propbank sections 2-21

still consider the preposition sense feature useful for SRL (at least in theory), only that the accuracy of the automatic tagger is not high enough.

5.3.3 Upper Bound for Preposition Sense Features in SRL

The experiments with the strong baseline showed that the automatically determined preposition sense feature failed to improve the SRL accuracy. This could either be because the accuracy of the automatic tagger is too low or because the information of the sense feature is already encoded in some of the other features. The goal of this experiment is to establish an upper bound for the leverage of the preposition sense feature, i.e. to determine the maximum effect the feature could possibly have.

Let us assume the existence of an oracle function that can disambiguate preposition sense with perfect accuracy. If the oracle failed to improve the SRL accuracy, it would mean that the information is already encoded in other features and would render the preposition sense feature useless. The oracle function can be simulated by directly using the gold standard preposition sense as a feature.

Because the preposition gold standard covers only three of the twenty sections of Propbank training data, the model can only be trained on this restricted training set. The preposition gold standard covers sections 2-4 which contain 5275 sentences. Although the training data is limited, it still allows to train competitive SRL models. [Pradhan et al.2005] have shown that the benefit of using more training data diminishes after a few thousand training instances. The accuracy of the SRL strong baseline model which is trained on three sections is only 3.89% lower compared to the model which is trained on twenty sections.

The gold annotated sense for the seven top prepositions is added as a feature and the model is re-trained and re-tested. The detailed scores of the experiment are shown in table 5.8. Note that this upper bound experiment is not conducted with the function tag sense, because that would mean giving gold SRL labels as input to the classifier.

The classification accuracy with the fine-grained gold sense feature is 0.65% higher than the strong baseline model. For coarse-grained sense, the improvement is 0.54%. Locative adjunctive roles (Argm-loc) improve about 7% and temporal roles (Argm-tmp) about 3% in F1 measure. Although the effect of the gold sense feature on the overall score is marginal, it is still satisfactory to see a decent improvement for the roles Argm-loc and Argm-tmp.

The experiment shows that the true preposition sense is helpful to classify temporal and locative semantic roles, even in a state-of-the-art model.

5.4 Classifier Combination for SRL

The experiments with the maxent SRL classifier showed that the true preposition sense can be a valuable feature for SRL, but that the automatically disambiguated sense is not accurate enough to improve a state-of-the-art SRL system when it is added as a feature. At the same time, we have seen in section 5.2.2 that the function tag of an adjunctive argument can be disambiguated with a high accuracy of over 80%.

Semantic Role	**Test Instances**	**Baseline**	**+Fine (gold)**	**+Coarse (gold)**
Overall	14508	87.89(A)	88.54(A)	88.43(A)
ARG0	3810	93.61	93.79	93.54
ARG1	5467	92.09	92.33	92.09
ARG2	1140	75.41	76.19	76.09
ARG3	177	68.75	68.39	67.71
ARG4	103	75.73	78.82	79.19
ARG5	5	88.89	88.89	61.54
ARGM-ADV	509	70.96	70.61	70.99
ARGM-CAU	77	72.86	67.61	73.24
ARGM-DIR	86	61.35	58.68	58.02
ARGM-DIS	319	87.04	87.14	87.50
ARGM-EXT	35	57.14	60.38	57.63
ARGM-LOC	384	69.74	77.42	76.52
ARGM-MNR	352	57.54	58.63	61.56
ARGM-MOD	554	99.46	99.46	99.28
ARGM-NEG	229	99.34	99.12	99.13
ARGM-PNC	115	60.91	60.63	60.44
ARGM-PRD	5	0.00	0.00	0.00
ARGM-REC	2	0.00	0.00	0.00
ARGM-TMP	1139	86.00	89.60	89.24

Table 5.8: Detailed scores for the strong baseline model and gold sense feature on the SRL classification task on Propbank section 23, trained on Propbank sections 2-4

I experiment with a classifier combination model to merge the output of the function tag WSD classifier and the strong baseline SRL classifier to achieve better classification results for adjunctive arguments. The combination model is implemented using a decision tree classifier that was trained on the predictions of both classifiers on the training sections 2 to 21.

The combination model is tested in two different settings: one where the distinction between adjunctive and non-adjunctive arguments is already known and one where the distinction is made automatically by a classifier (adjunctive-PP classification). All instances, that were identified as adjunctive prepositional phrases are classified by the combination model, based on the output of the two individual classifiers. All other instances are classified as before. The system is evaluated over all semantic roles, adjunctive and non-adjunctive in Propbank section 23.

The results for the adjunctive-PP classification step and the combination model can be seen in table 5.9. For easy reference, I also include the accuracy for the strong baseline model without combination and the accuracy of the function tag WSD classifier. The combination model shows a modest performance increase of 1.54% over the strong base model in the oracle set up. In the automatic setup, the increase is marginal (0.05%).

I again report the detailed scores to see if the improvement for location and temporal roles would be higher. The F1 measure for Argm-loc increased by 0.62% in the automatic setting and 2.73% in the oracle setting. For Argm-tmp the F1 measure dropped slightly by 0.02% and 0.1%. The complete detailed scores are shown in table 5.10.

The classifier combination model is an interesting approach to combine information from WSD and SRL, however the model only shows improvement for the location

	Classification	**Adjunctive-PP classification**		
	A	P	R	F1
Baseline SRL	91.78	-	-	-
Function tag WSD	84.5	-	-	-
Argm-PP classification	-	89.78	92.08	90.92
Combination (auto)	91.83	-	-	-
Combination (oracle)	93.32	-	-	-

Table 5.9: Scores for the combination model on the SRL classification task on Propbank section 23

Semantic Role	**Test Instances**	**Baseline**	**Combination (oracle)**	**Combination (auto)**
Overall	14508	91.78(A)	93.32(A)	91.83(A)
ARG0	3810	95.74	95.77	95.73
ARG1	5467	94.83	95.18	94.91
ARG2	1140	85.10	87.40	85.42
ARG3	177	78.08	82.65	78.88
ARG4	103	84.76	90.00	84.76
ARG5	5	90.91	90.91	90.91
ARGM-ADV	509	74.90	73.98	75.15
ARGM-CAU	77	78.32	80.27	79.72
ARGM-DIR	86	78.31	77.89	75.28
ARGM-DIS	319	89.16	90.03	89.47
ARGM-EXT	35	64.41	61.29	59.02
ARGM-LOC	384	81.26	83.99	81.88
ARGM-MNR	352	69.05	72.05	70.04
ARGM-MOD	554	99.64	99.46	99.64
ARGM-NEG	229	99.35	99.35	99.35
ARGM-PNC	115	67.58	73.19	65.79
ARGM-PRD	5	28.57	0.00	33.33
ARGM-REC	2	0.00	0.00	0.00
ARGM-TMP	1139	91.75	91.73	91.65

Table 5.10: Detailed scores for the combination model on the SRL classification task on Propbank section 23, trained on Propbank sections 2-21

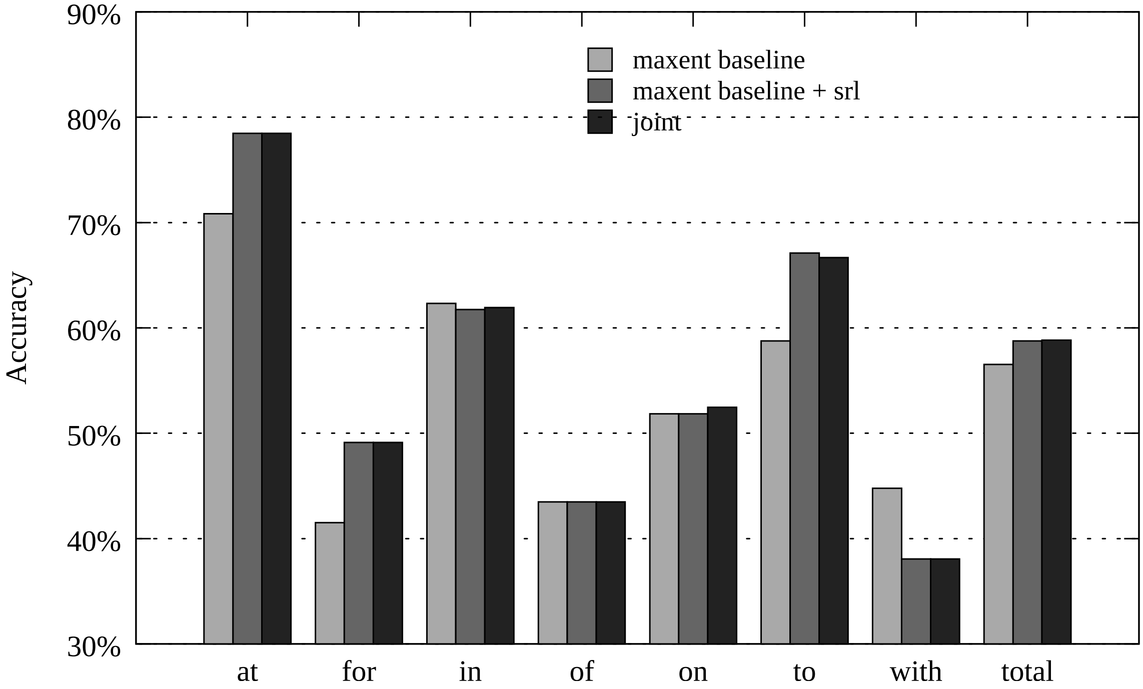


Figure 5.6: Coarse-grained classification accuracy of the maxent WSD models and joint model on prepositions in Propbank section 23

argument and the improvement is only significant when the distinction between adjunctive and core arguments is known.

A major drawback of the combination model are the additional classification steps. First, arguments have to separated from non-arguments, then adjunctive arguments from core arguments, then function tags and semantic roles are classified and finally yet another classification step is needed to combine the classifiers. Each additional classifier adds noise to the following steps, thus diminishing the improvement of the combination model.

5.5 Joint Learning Semantic Roles and Preposition Sense

In this section, I present the results for a joint learning approach to semantic roles and preposition sense. Instead solving the two tasks in a pipeline of classifiers, the model seeks to find the semantic role and preposition sense pair that maximizes the joint probability over both labels,

The performance of the joint learning model for the WSD and SRL task is evaluated on the annotated prepositional phrases in test section 23. The performance is compared with the performance of an independent baseline model and an extended baseline model that use the prediction of the other task as a feature. Note that in this experiment, the models are evaluated on prepositional phrases only.

Figure 5.6 shows the coarse-grained classification accuracy of the WSD models for each of the seven prepositions on the test instances in Propbank section 23. The detailed scores are given in table 5.11. The results show that the extended model and the joint model perform almost identically, the joint model performing marginally

Preposition	Test Instances	Baseline	Baseline+SRL	Joint
at	144	70.83	78.47*	78.47*
for	171	41.52	49.12*	49.12*
in	507	62.33	61.74	61.93
of	46	43.48	43.48	43.48
on	162	51.85	51.85	52.47
to	228	58.77	67.11*	66.67*
with	134	44.78	38.06	38.06
Total	1392	56.54	58.76*	58.84*

Table 5.11: Accuracy of the maxent WSD model, extended WSD model and joint model on the WSD task on prepositions in Propbank section 23, statistically significantly improved results are marked with (*)

Semantic Role	Test Instances	Baseline	Baseline+WSD	Joint
Overall	1389	71.71(A)	69.47(A)	72.14(A)
ARG0	13	47.62	13.33	42.11
ARG1	166	68.12	65.85	66.12
ARG2	295	78.06	78.68	79.03*
ARG3	44	68.89	53.33	70.33
ARG4	86	87.06	86.90	87.06
ARGM-ADV	40	31.43	30.00	29.73
ARGM-CAU	8	0.00	0.00	0.00
ARGM-DIR	9	14.29	16.67	15.38
ARGM-DIS	20	57.89	24.00	55.56
ARGM-EXT	19	88.24	88.24	88.24
ARGM-LOC	286	72.88	71.54	74.27*
ARGM-MNR	91	41.38	42.39	38.57
ARGM-PNC	37	34.62	39.39	38.57
ARGM-REC	1	0.00	0.00	0.00
ARGM-TMP	274	81.87	79.43	83.24*

Table 5.12: Detailed scores for the baseline, extended baseline and the joint model on the SRL task on Propbank section 23, trained on Propbank sections 2-4, statistically significant results are marked with (*)

better in the overall score. Both models outperform the baseline classifier in three of the seven prepositions. The Student's t-test shows that the improvement is statistically significant ($p < 0.05$) in these three cases and in the total score.

The fact that the extended model and the joint model show almost identical performance suggests that the SRL feature has a strong impact on the WSD model but that in the opposite direction, the WSD component has less effect on the SRL model.

For the SRL task, I report the accuracy over all annotated prepositional phrases in Propbank section 23 and the F1 measure for individual semantic roles. Figure 5.7 shows the results for the semantic roles Argm-loc and Argm-tmp and the overall accuracy. The detailed results are listed in table 5.12. The joint model shows a performance increase by 0.43% over the baseline on the overall accuracy. Adding the preposition sense as a feature, on the other hand, significantly lowers the accuracy by over 2%. The drop in accuracy is similar to the results we observed in section 5.3.2. For the roles Argm-loc and Argm-tmp, the joint model improves the F1 measure by about 1.4% respectively. The improvement is statistically significant

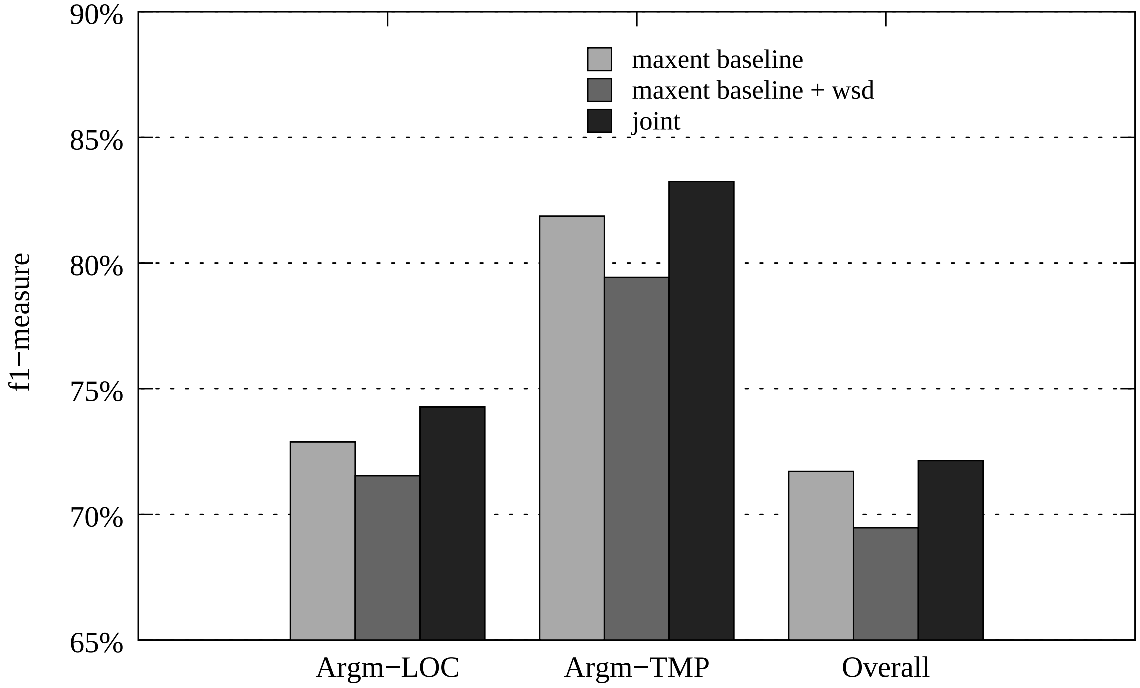


Figure 5.7: F1 measure of the baseline, extended and joint model for Argm-loc, Argm-tmp and overall accuracy on the SRL classification task for prepositional phrases in Propbank section 23, trained on Propbank sections 2-4

($p \leq 0.05$, Student's t-test). Simply adding the preposition sense as a feature again lowers the F1 measure.

The reason for the poor performance of the extended model is that the sense feature contains a high degree of noise. We have seen that the sense is strongly correlated with the semantic role, thus it receives high confidence from the maxent classifier during training. During testing, the supposedly helpful sense label is only correct in about half of the cases, thus causing the SRL classifier to make more erroneous predictions. The results suggest that a joint model is more robust to noisy features than a pipeline of sequential classifiers.

5.6 Discussion and Analysis

In most of the experiments, the preposition sense feature failed to significantly improve the SRL classification accuracy. The two exceptions are the weak baseline model and the oracle model, where preposition senses are disambiguated with perfect accuracy.

In the weak baseline model, the preposition sense features improves the accuracy, because the baseline features do not discriminate well between different prepositional phrases. The only constituent specific features in the weak model are the head word and the phrase type. The head word of a prepositional phrase is the preposition itself, so `in the kitchen` and `in the morning` both have the same head word `in`. The two phrases also have the same phrase type and can appear in the same position in the parse tree. Adding the sense of the head preposition helps the model to find the semantic role of the argument, because the sense information is not encoded in any other feature.

In the strong baseline model, there are more constituent specific features, in particular the first and last word of the constituent, the head word and POS of the rightmost noun phrase and temporal cue words. These features already encode information about the preposition sense. Especially the rightmost noun phrase is a strong hint for both the semantic role and the preposition sense. If the training data is not too sparse, the classifier will probably have seen the words `kitchen` and `morning` as part of the roles Argm-loc and Argm-tmp in the training data already (I found that the only occurrence of `kitchen` as part of an adjunctive argument in the Propbank training data was indeed labeled as Argm-loc). The constituent specific SRL features overlap with the WSD features. The average length of a prepositional phrase in Propbank is 2.3 words. The window size for surrounding POS and local collocation features is five and seven respectively. Therefore, the first and last word and their POS as well as the rightmost noun phrase will usually appear as features in the WSD classifier as well. The amount of "new" information that is added by the WSD classifier in the strong baseline model is not as large as in the weak baseline model, because the information is partly encoded in other features already.

On the other hand, the gold preposition sense is still able to improve the SRL classification accuracy. This might suggest that some prepositional phrases that are misclassified by the SRL classifier require deep world knowledge that is hard to learn for a statistical classifier. Consider the following sentence:

> "One of the things that worry me is the monetary welfare...," said Lawrence Kudlow, a Bear, Stearns & Co. economist, [*on* ABC's "This Week"].
>
> (wsj_2384.s13)

The prepositional phrase was misclassified by the strong baseline model as Argm-tmp, because the last word `week` is a strong hint for a temporal role. The correct semantic role label would have been Argm-loc. To classify this label correctly, a human judge might rely on logical inference about the described situation, instead of merely relying on surrounding words: ABC is a television channel and "This Week" must therefore be the name of a television show and not a time specification. This kind of inference is possible for a human annotator (who is roughly familiar with American TV channels), but not for a statistical classifier.

Other methods to incorporate the preposition sense into the SRL system showed ambivalent results. Classifier combination failed to improve the overall performance, because too many consecutive classification steps are involved. I found several examples where the combination model successfully corrected the SRL classifier when important key words were hidden *inside* the prepositional phrase. The following prepositional phrase, for example, was misclassified by the strong baseline SRL model as Argm-loc when the correct semantic role would have been Argm-tmp.

> Rightly or wrongly, many giant institutional investors appear to be fighting the latest war by applying the lessons they learned [in the October 1987 crash]:....
>
> (wsj_2381.s21)

When looking into the training data, I found that the predominant function tag for prepositional phrases of the phrase `in ...crash` was indeed Argm-loc (seven occurrences for Argm-loc and three for Argm-tmp). The SRL classifier extracts the word `crash` as a feature (last word and right most NP child), but does not include the word `October` in any feature. The unigram `October` just behind the preposition is a strong hint for Argm-tmp. I found six instances of Argm-tmp that had the unigram `October` within a three token window following the preposition `in`, compared to one instance of Argm-loc.

In this case, the function tag WSD classifier assigned the correct class and corrected the SRL model. However, the additional classification step for splitting the instances in adjunctive and core arguments diminishes most of the improvement we get through combination.

Finally, the joint learning model for semantic roles and preposition sense showed decent results when classifying prepositional phrases. The model significantly boosts the accuracy on the WSD task and gives a statistical significant improvement for classifying the semantic roles Argm-loc and Argm-tmp. The following instance is an example where the joint model successfully corrected the SRL model. The constituent and the preposition that are considered are marked by brackets.

> Fidelity [on Saturday] opened its 54 walk-in investor centers across the country.
>
> (wsj_2306.s39)

The most likely semantic role according to the baseline SRL model would be Argm-loc with a probability of

$$P(l = \textsc{Argm-loc} | \Phi(t, p, v)) = 0.434$$

However, the joint probability of the role Argm-loc and the most likely sense of `on` is lower than the joint probability for the role Argm-tmp. The final prediction is (Argm-tmp, on_8) which is also the correct answer.

$$\begin{aligned} P(\text{l=}\textsc{Argm-loc}|\Phi(t,p,v)) \quad &\times \quad P(\text{s=on\_8}|\Psi(\text{c,}\textsc{Argm-loc})) = 0.122 \\ P(\text{l=}\textsc{Argm-tmp}|\Phi(t,p,v)) \quad &\times \quad P(\text{s=on\_8}|\Psi(\text{c,}\textsc{Argm-tmp})) = 0.304 \end{aligned}$$

In this example, the temporal meaning of the preposition phrase can be guessed just from the two words `on Saturday` without looking at other features from the parse tree. The WSD classifier, which uses local collocation features from a narrow window, can determine the sense of the preposition more accurately.

5.7 Summary

My experiments show that the preposition sense can be a valuable knowledge source for SRL classification, but not for SRL identification. The impact of the preposition knowledge source depends on the strength of the SRL classification model and the quality of the sense annotation. When automatically disambiguated senses are incorporated in a baseline SRL classification model, the feature significantly improves the classification accuracy, but in a more complex model, the feature fails to improve the performance. The reason is that many of the features that are used by the WSD classifier are already present in the SRL model and the classifier only adds little new information to the classification model. The automatically tagged sense feature is not very accurate and can even cause the SRL classifier to make more erroneous predictions.

On the other hand, experiments with gold standard preposition sense show that the true preposition sense is still able to improve the classification accuracy of locative and temporal adjunctive arguments. This suggests that a more accurate preposition WSD classifier would be needed to increase the leverage of preposition sense features for SRL.

Finally, my experiments with a joint model show decent improvement over competitive individual models. The joint model is more robust to noise in the preposition sense feature and shows that different semantic classification tasks for prepositional phrases can benefit from joint inference methods. To the best of my knowledge, this is the first system that learns semantic roles and preposition senses in a joint model.

6. Summary

on In this thesis, I investigated the use of preposition sense for the semantic role labeling task. I adapted a classifier from the literature for preposition sense disambiguation. The system outperformed existing state-of-the-art systems on the SemEval 2007 preposition WSD data set. The classifier was used to automatically tag prepositions in the Propbank corpus for the SRL task. The results of my experiments showed a strong accuracy drop, due to cross-domain problems. Consequently, I manually annotated the sense for the seven most frequent prepositions in four sections of the Propbank corpus to get domain specific training and test instances. The domain specific training data significantly improved the WSD classification accuracy.

Following the WSD experiments, I empirically studied the leverage of the preposition sense for semantic role labeling. Adding the preposition sense as a feature to a baseline SRL model showed that the sense can be a valuable knowledge source for the classification step, but not for identification. However, the utility of the preposition feature depended on the strength of the SRL features and the accuracy of the sense annotation. While the feature significantly improved the classification accuracy in the baseline SRL model, the feature fails to improve the performance of a more sophisticated model.

I experimented with a classifier combination approach to combine SRL and WSD, but the improvement was marginal, because too many sequential classification steps were involved.

Finally, I proposed a probabilistic model to jointly classify semantic roles of prepositional phrases and the sense of the associated preposition. The joint model showed a decent improvement over competitive, independent models for each sub-task. As I only used the senses of the seven most frequent prepositions, more leverage could be gained by including more prepositions or complex phrasal prepositions into the joint model. To overcome the need to manually label the training data, it would be interesting to investigate methods to create the training data in an unsupervised or semi-supervised fashion. I leave this for future work.

Bibliography

[Andrew et al.2004] Galen Andrew, Trond Grenager, and Cristopher Manning. 2004. Verb Sense and Subcategorization: Using Joint Inference to Improve Performance on Complementary Tasks. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 150–157.

[Baker et al.1998] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING/ACL 1998)*.

[Brill and Wu1998] Eric Brill and Jun Wu. 1998. Classifier Combination for Improved Lexical Disambiguation. In *Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics (COLING/ACL 1998)*, pages 10–14.

[Carreras and Màrquez2004] Xavier Carreras and Lluís Màrquez. 2004. Introduction to the conll-2004 shared task: Semantic role labeling. In *Proceedings of the 8th Conference on Computational Natural Language Learning (CoNLL 2004)*, pages 89–97.

[Carreras and Màrquez2005] Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL 2005)*, pages 152–164.

[Charniak2001] Eugene Charniak. 2001. Immediate-head parsing for language models. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001)*, pages 124–131.

[Cohn and Blunson2005] Trevor Cohn and Philip Blunson. 2005. Semantic role labeling with tree conditional random fields. In *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL 2005)*, pages 169–172.

[Collins2003] Michael Collins. 2003. Head-Driven Statistical Models for Natural Language Parsing. *Computational Linguistics*, 29(4):589–637.

[Dang and Palmer2005] Hoa Trang Dang and Martha Palmer. 2005. The Role of Semantic Roles in Disambiguating Verb Senses. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 42–49.

[Dowty1991] David Dowty. 1991. Thematic Proto-roles and Argument Selection. *Language*, 67(3):547–619.

[Fellbaum1998] Christiane Fellbaum. 1998. *WordNet: an Electronic Lexical Database.* MIT Press Cambridge, MA, USA.

[Foley and Valin1984] Wa Foley and Robert D.vab Valin. 1984. Functional Syntax and Universal Grammar. *Cambridge Studies in Linguistics London.*

[Gildea and Jurafsky2002] Daniel Gildea and Daniel Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3):245–288.

[Gruber1970] Jeffrey S. Gruber. 1970. *Studies in Lexical Relations.* Indiana University Linguistics Club.

[Hirst1987] Graeme Hirst. 1987. *Semantic Interpretation and the Resolution of Ambiguity.* Cambridge University Press.

[J.Fillmore1967] Charles J.Fillmore. 1967. The Case for Case. In *Proceedings of the Texas Symposium on Language Universals.*

[Jiang and Ng2006] Zheng Ping Jiang and Hwee Tou Ng. 2006. Semantic Role Labeling of NomBank: A Maximum Entropy Approach. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*, pages 138–145.

[Jurafsky and Martin2006] Daniel Jurafsky and James H. Martin. 2006. *Speech and Language Processing.* Prentice-Hall, Inc. Upper Saddle River, NJ, USA.

[Lee and Ng2002] Yoonk Keok Lee and Hwee Tou Ng. 2002. An Empirical Evaluation of Knowledge Sources and Learning Algorithms for Word Sense Disambiguation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 41–48.

[Litkowski and Hargraves2005] Kent Litkowski and Orin Hargraves. 2005. The Preposition Project. In *Proceedings of the 2nd ACL-SIGSEM Workshop on Prepositions*, pages 171–179.

[Litkowski and Hargraves2007] Kent Litkowski and Orin Hargraves. 2007. SemEval-2007 Task 06: Word-Sense Disambiguation of Prepositions. *SemEval-2007: 4th International Workshop on Semantic Evaluations.*

[Litkowski2004] Kent Litkowski. 2004. Senseval-3 task: Automatic labeling of semantic roles. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 9–12.

[Marcus et al.1994] Mitchell P. Marcus, Grace Kim, Mary A. Marcinkiewicz, Robert MacIntyre, Ann Bies, MarkcFerguson, Karen Katz, and Britta Schasberger. 1994. The Penn Treebank: Annotating Predicate Argument Structure. In *ARPA Human Language Technology Workshop*, pages 114–119.

[Meyers et al.2004] Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. Annotating noun argument structure for NomBank. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004).*

[O'Hara and Wiebe2003] Tim O'Hara and Janyce Wiebe. 2003. Preposition Semantic Classification via Treebank and FrameNet. In *Proceedings of the 7th Conference on Natural Language Learning (CoNLL 2003)*, pages 79–86.

[Palmer et al.2001] Martha Palmer, Hoa Trang Dang, and Joseph Rosenzweig. 2001. Sense Tagging the Penn Treebank. In *Proceedings of the 2nd Language Resources and Evaluation Conference.*

[Palmer et al.2005] Martha Palmer, Paul Kingsbury, and Daniel Gildea. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

[Pradhan et al.2005] Sameer Pradhan, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H. Martin, and Daniel Jurafsky. 2005. Support Vector Learning for Semantic Argument Classification. *Machine Learning*, 60(1):11–39.

[Punyakanok et al.2004] Vasin Punyakanok, Dan Roth, Wen tau Yih, and Dav Zimak. 2004. Semantic Role Labeling via Integer Linear Programming Inference. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004).*

[Quirk1985] Randoplh Quirk. 1985. *A Comprehensive Grammar of the English Language.* Longman.

[Ratnaparkhi1996] Adwait Ratnaparkhi. 1996. A Maximum Entropy Model for Part-of-Speech Tagging. In *Proceedings of the 1996 Conference on Empirical Methods in Natural Language Processing (EMNLP 1996)*, pages 133–142.

[Soanes and Stevenson2003] Catherine Soanes and Angus Stevenson. 2003. *Oxford Dictionary of English.* Oxford University Press New York.

[Stevenson and Wilks2001] Mark Stevenson and Yorick Wilks. 2001. The Interaction of Knowledge Sources in Word Sense Disambiguation. *Computational Linguistics*, 27(3):321–349.

[Surdeanu et al.2003] Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using Predicate-argument Structures for Information Extraction. In *Proceedings of the 41rd Annual Meeting of the Association for Computational Linguistics (ACL 2003).*

[Tjong et al.2005] Erik F. Tjong, Kim Sang, and Fien De Meulder. 2005. Applying Spelling Error Correction Techniques for Improving Semantic Role Labeling. In *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL 2005).*

[Toutanova et al.2005] Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. 2005. Joint Learning Improves Semantic Role Labeling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 589–596.

[Witten and Frank2005] Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann San Francisco, CA, USA.

[Xue and Palmer2003] Nianwen Xue and Martha Palmer. 2003. Annotating the Propositions in the Penn Chinese Treebank. In *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing*.

[Xue and Palmer2004] Nianwen Xue and Martha Palmer. 2004. Calibrating Features for Semantic Role Labeling. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 88–94.

[Ye and Baldwin2006] Patrick Ye and Timothy Baldwin. 2006. Semantic Role Labeling of Prepositional Phrases. *ACM Transactions on Asian Language Information Processing (TALIP)*, 5(3):228–244.

[Ye and Baldwin2007] Patrick Ye and Timothy Baldwin. 2007. MELB-YB: Preposition Sense Disambiguation Using Rich Semantic Features. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval 2007)*, pages 241–244.