# Using Lexical and Compositional Semantics to Improve HPSG Parse Selection

Zinaida Pozen

A thesis
submitted in partial fulfillment of the
requirements for the degree of

Master of Science

University of Washington

2013

Reading Committee:

Emily Bender, Chair

Francis Bond

Program Authorized to Offer Degree:
Computational Linguistics

**Abstract**

Using Lexical and Compositional Semantics
to Improve HPSG Parse Selection

Chair of the Supervisory Committee:
Dr. Emily Bender
University of Washington

Accurate parse ranking is essential for deep linguistic processing applications and is one of the classic problems for academic research in NLP. Despite significant advances, there remains a big need for improvement, especially for domains where gold-standard training data is scarce or unavailable. An overwhelming majority of parse ranking methods today rely on modeling syntactic derivation trees. At the same time, parsers that output semantic representations in addition to syntactic derivations (like the monostratal DELPH-IN HPSG parsers) offer an alternative structure for training the ranking model, which could be further combined with the baseline syntactic model score for re-ranking. This thesis proposes a method for ranking the semantic sentence representations, taking advantage of compositional and lexical semantics. The methodology does not require sense-disambiguated data, and therefore can be adopted without requiring a solution for word sense disambiguation. The approach was evaluated in the context of HPSG parse disambiguation for two different domains, as well as in a cross-domain setting, yielding relative error rate reduction of 11.36% for top-10 parse selection compared to the baseline syntactic derivation-based parse ranking model, and a standalone ranking accuracy approaching the accuracy of the baseline syntactic model in the best setup.

## TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

## ACKNOWLEDGMENTS

I am deeply grateful to my advisers, Prof. Emily Bender at the University of Washington, and Prof. Francis Bond from Nanyang Technological University, whose depth of knowledge and experience, scientific curiosity and personal warmth have made this work possible. They have shared generously of their busy time, paid genuine attention to all aspects of the work, fully engaged in discussions and gave me so much helpful feedback. Thank you. I have also been very lucky to work with Prof. Katrin Kirchhoff at the University of Washington EE CSLI Lab, and had an opportunity to discuss several aspects of this research with her, and get her helpful advise. I am full of appreciation to the University of Washington CLMS Program faculty (Profs. Emily Bender, Gina Levow, Scott Farrar and Fei Xia) for designing the curriculum relevant to today's computational linguistics research and applications, and teaching the classes in an engaged, warm and accessible manner that cares about the students' success. I have received wonderful support from David Brodbeck, Mike Furr and Joyce Parvi at the UW Linguistics department both as a student and as a TA/RA. I had many helpful discussions with fellow students while working on my thesis: Amittai Axelrod, Jim White, Prescott Klassen, Lauren Wang, Glenn Slayden, Michael Goodman, Sanghoun Song, Spencer Rarrick, Joshua Crowgley, and members of Luke Zettlemoyer's summer 2012 Semantics Reading group. Special gratitude goes to Woodley Packard, who not only discussed the methodology in depth, but went over the data and the numbers, tried to reproduce them, and helped uncover several bugs in my code. Attending the DELPH-IN Summit in 2012 was a great experience that helped me go from a vague idea of wanting to do something with lexical semantics and HPSG, to the practical task with concrete evaluation criteria. Special thanks to Yi Zhang for helping me set up and run an early round of experimentation on his Jigsaw system. I have used ShareLaTeX for creating this document, and have been tremendously impressed with the

two-person team who have been putting in multiple improvements, and personally answering support questions. Finally, huge, huge thank you to my loving and patient family. Especially to Eddy for the wonderful illustrations: the goat-sawing cats are forever seared in many brains now.

## DEDICATION

To meaning!

# Chapter 1
# INTRODUCTION

This thesis presents a methodology aiming to combine structural and lexical semantic information to build a unified semantic model, and a series of experiments evaluating this approach for the task of parse selection. This work is performed in the context of the Head-Driven Phrase Structure Grammar (HPSG; Pollard & Sag, 1994), Minimal Recursion Semantics (MRS; Copestake et al., 2005), the English Resource Grammar (ERG; Flickinger, 2000) and WordNet (Fellbaum, 1998).

## *1.1 Motivation*

This effort was shaped by two considerations. To a large extent, it was driven by an entirely practical need to improve automatic parse selection precision. A variety of clever syntactic features go into producing the current syntactic parse disambiguation probabilistic models, yielding fairly high precision results: up to 76% full sentence accuracy has been reported for HPSG parsers (Toutanova et al., 2005). However, it remains a very desirable goal to improve these scores for producing high-quality automatically labeled data, especially since in real setups, when in-domain training data is not available, and these scores often end up quite a bit lower (around 40% is fairly common). Despite the fact that HPSG parsers produce both syntactic and semantic output, current implemented disambiguation models are not utilizing the semantic information. However, Fujita et al. (2007) demonstrated that using semantics, both structural and lexical, helps, improving parse disambiguation accuracy by almost 6% absolute for Japanese.

The second source of motivation is neatly summarized by this Philip Resnik's quote:"My ultimate focus is a basic question about language: what combinations of words make sense, semantically?" (Resnik, 1998, p. 240) That question remains unanswered and rarely addressed to this day. Modeling lexical semantics is not a very popular topic in current

computational linguistics, with the exception of word sense disambiguation tasks, boosted by the Senseval exercises (Kilgarrif, 1998; Edmonds & Cotton, 2001). The majority of the current research methods build upon surface N-grams, or syntactic categories. Some of the attempts, both early (Bikel, 2000) and recent (MacKinlay et al., 2012), to incorporate lexical semantic information to enhance parse selection precision have not demonstrated improvements and likely discouraged that research direction.

Another factor slowing down this research direction is the lack of proven, agreed-upon semantic categories, similar to broad parts of speech in syntax. Many motivating examples fueling lexical semantic research in the 80s and the 90s were drawing from the transfer-based machine translation scenarios (e.g., the analysis of movement verbs in English and Spanish in Jackendoff (1992)). With surprisingly good performance of purely surface-oriented statistical machine translation systems, the need for lexical semantics appeared to have been diminished. Lexical information, even over very large corpora, introduces sparsity problems and is therefore often avoided in favor of de-lexicalized models heavily relying on syntactic regularities (see, for example, Zhang & Krieger (2011)).

The intuition behind my approach is that different lexical semantic classes (as defined by lexical taxonomies) have distinct distributional properties, especially pronounced with respect to the sentence structural or frame semantics. For example, humans are much more likely than artifacts to serve as first arguments to verbs of communication. This fact can be exploited to build semantically generalized language models that back off from lexically sparse open class words to a small set of high-level semantic categories.

### *1.2 Motivating Example*

Consider the simple sentence (1), with some of the many possible interpretations paraphrased.

(1) *I saw a kid with a cat.*

   a. *I saw [with my eyes] a child who was together with a cat.* (Fig 1.1)

   b. *I, using a cat, saw [with my eyes] a child.* (Fig 1.2)

   c. *I cut with a saw a child who was together with a cat.* (Fig 1.3)

d. *I, together with a cat, cut with a saw a young goat.* (Fig 1.4)

Figures 1.1 through 1.4, show the HPSG parse trees and corresponding Elementary Dependency (Oepen & Lønning, 2006) semantic structures paired with possible interpretive illustrations.[1] The semantic analysis is provided by the ERG (Flickinger, 2000) version 1111 on-line demo.[2]

(1) exhibits both lexical and structural ambiguity, as shown in (1a–1d) and Figures 1.1 through 1.4. One type of lexical ambiguity in these examples comes from the verbs *to see* and *to saw* (not ordinarily even considered homonymous!) exhibiting the same surface form when *to see* is in the past tense, and *to saw* in the present, non-third person singular. These verbs are of course represented by different lexical entries in both WordNet and the ERG lexicons.

The second kind of lexical ambiguity is between two senses of *kid* - one meaning "child", and another meaning "young goat". This ambiguity is reflected in WordNet, but not reflected in the ERG lexicon: both the ERG *kid*s are modeled by the same ERG lexical item, because they exhibit the same syntactic behavior and open-class semantics characteristic of common nouns.

Finally, *with* is also lexically ambiguous between comitative (together with) and instrumental (using) meanings. 1.2 illustrates the instrumental use of *with*. Neither the ERG nor WordNet lexicon model different meanings of *with*, because WordNet only includes open-class words (nouns, verbs, adjectives and adverbs), whereas the ERG, again, does not multiply lexical entries if they behave the same way syntactically.

The syntactic ambiguity comes from PP attachment. Figures 1.1 and 1.3 reflect a low PP attachment, where *kid* forms a constituent with *cat*, while parses in 1.2 and 1.4 have a high PP attachment, grouping the head verb (*see* or *saw*) with *cat*. The semantic predicate structure mirrors this attachment ambiguity in the argument structure for *with*: whether its first and second arguments bring together the *kid* and the *cat*, or the *seeing/sawing* with

---

[1] I asked *my* kid to draw these.

[2] The on-line demo is available at `http://erg.delph-in.net/logon`.

the *cat*.[3]

The current parse ranker trained on the syntactic features prefers the reading reflected in 1.2 where *seeing* is happening with the cat. However, the most neutral and likely interpretation of this sentence is, of course, 1.1. It seems like the best hope to select it as the preferred parse would be to model the semantics of the sentence. Seeing with animals is just not something we frequently do. More formally (and assuming we can make use of lexical semantic categories), the predicate *with* is not likely to take the verb of perception such as *see* and a noun signifying an animal (*cat*) as its two arguments.

In the rest of the thesis, I will present my attempt to train a model that would reflect these intuitions, and improve on syntax-based parse selection results.

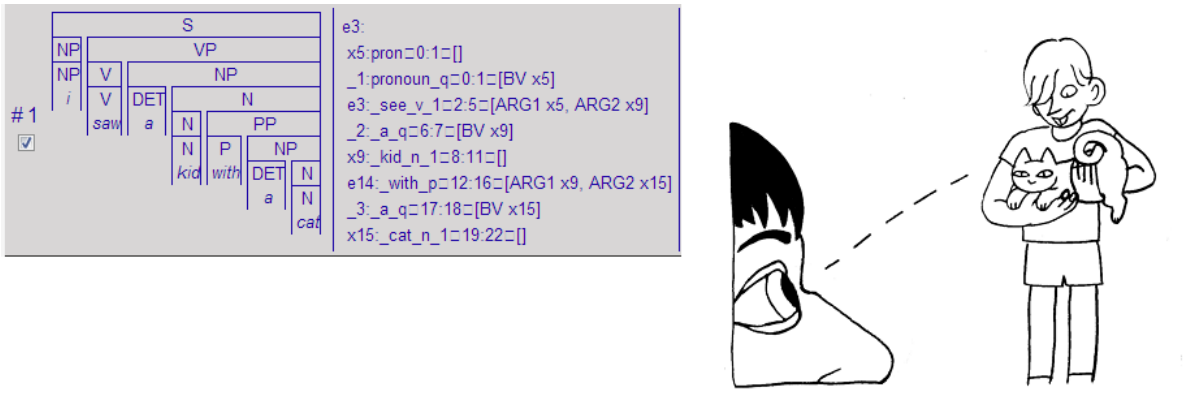


Figure 1.1: I saw a kid with a cat — 1a

### *1.3 Structure of the Thesis*

The goal of this chapter has been to motivate the work from a high level perspective. The rest of the thesis is structured as follows: Chapter 2 gives a brief overview of the theories and applications in the context of which these experiments were carried out. Chapter 3 surveys the state of the art for incorporating lexical semantic features in parsing. Chapter 4

[3]It is worth noting that these illustrations are of course just a subset of the possible interpretations and sense combinations for the sentence.

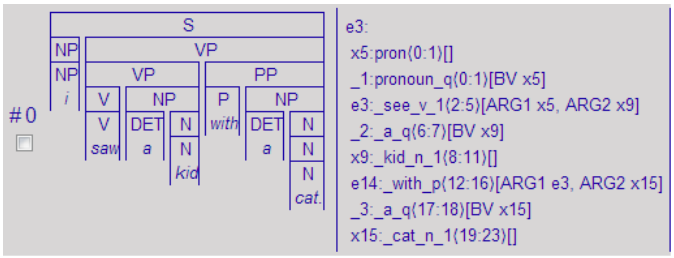


Figure 1.2: I saw a kid with a cat — 1b

describes the methodology employed in my experiments. In Chapter 5, I present the experimental results, discuss the results, and perform some error analysis. Finally, Chapter 6 presents the conclusions, some musings about what this all means, unanswered questions and ideas for future work.

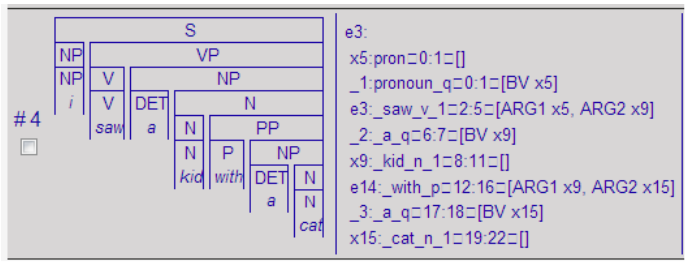


Figure 1.3: I saw a kid with a cat — 1c

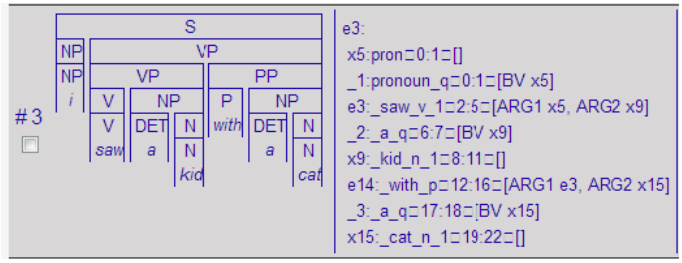


Figure 1.4: I saw a kid with a cat — 1d

# Chapter 2

# BACKGROUND

This chapter surveys theories, formalisms and resources that serve as the foundation for my thesis. I briefly outline HPSG and Minimal Recursion Semantics and describe the current approach to parse selection for HPSG parsers. After introducing Princeton WordNet and WordNet semantic files, I talk about the English Resource Grammar, focusing on its lexicon and how it compares to WordNet. The chapter ends with a brief overview of the MALLET toolkit.

## *2.1 HPSG, MRS, ERG and DELPH-IN tools*

Head-Driven Phrase Structure Grammar (HPSG; Pollard & Sag, 1994; Sag et al., 2000) is a monostratal, unification-based theory of grammar. It is formally precise, supporting multiple parser implementations, such as LKB (Copestake, 2002), PET (Callmeier, 2000), ACE,[1] AGREE,[2] and ALE (Carpenter & Penn, 1994). Broad-coverage grammars based on the HPSG formalism exist for multiple languages, the English Resource Grammar (ERG; Flickinger, 2000) being the most mature among them. The Grammar Matrix project (Bender et al., 2002) in particular has served as a foundation for creating grammar implementations for a wide array of typologically-diverse languages by providing a set of libraries for describing phenomena based on careful cross-linguistic analysis. An international research collaboration, the Deep Linguistic Processing with HPSG Initiative, or DELPH-IN,[3] has been fostering research and development of open-source tools and grammars for HPSG.

One of the distinguishing features of DELPH-IN HPSG grammar formalism and implementations is that parsing produces not only syntactic phrase structure trees, but also Min-

[1] `http://sweaglesw.org/linguistics/ace`

[2] `http://moin.delph-in.net/AgreeTop`

[3] `http://www.delph-in.net`

imal Recursion Semantics (MRS; Copestake et al., 2005) representations of the structural semantics. The individual predications are contributed by the lexical items and grammar rules and combined into a representation that flattens the sentence semantics into a bag of *elementary predications*, or EPs, that are linked by variables connecting semantic roles across EPs. The representation is underspecified for quantifier scope, sidestepping the question of difficult but rarely practically necessary quantifier scope resolution. MRS also distinguishes the top-most EP for every sentence, corresponding to the most semantically prominent predication.

As a concrete example, let's examine again the semantic structure assigned to our earlier example sentence, *I saw a kid with a cat*, as shown in Figure 2.1. There are eight elementary predications in this sentence, with the top one corresponding to the *see*ing event.

$h_4$:pron($x_5$\{PERS *1*, NUM *sg*, PRONTYPE *std_pron*\}),
$h_6$:pronoun_q($x_5$, $h_7$, $h_8$),
$h_2$:_see_v_1($e_3$\{SF *prop*, TENSE *pres*, PROG -, PERF -\}, $x_5$, $x_9$\{PERS *3*, NUM *sg*, IND +\}),
$h_{10}$:_a_q($x_9$, $h_{12}$, $h_{11}$),
$h_{13}$:_kid_n_1($x_9$),
$h_{13}$:_with_p($e_{14}$\{SF *prop*, TENSE *untensed*\}, $x_9$, $x_{15}$\{PERS *3*, NUM *sg*, IND +\}),
$h_{16}$:_a_q($x_{15}$, $h_{18}$, $h_{17}$),
$h_{19}$:_cat_n_1($x_{15}$)
$h_{18} =_q h_{19}$, $h_{12} =_q h_{13}$, $h_7 =_q h_4$, $h_1 =_q h_2$

Figure 2.1: ERG MRS output for sentence *I saw a kid with a cat*, slightly simplified for illustration purposes.

Some of the predicates (such as *_kid_n_1* or *_see_v_1*) correspond to contentful surface words: these are called *real* predicates, or *realpreds*. Others are grammatical predicates (or *gpreds*): these are elements of lexical semantic generalization or decomposition, or posited as a contribution of grammar rules. One example of a semantic generalization *gpred* is *pron* predicate, posited for all pronouns, with *PERS*, *NUM* and *PRONTYPE* features to distinguish between different pronoun types. Other gpreds in the ERG include predicates

of negation, possession, location, time, part-of, etc. The set of gpreds is similar, but not the same, across different DELPH-IN grammars. There is an ongoing effort to harmonize them.

There is not a one-to-one correspondence between the syntactic leaf nodes' tokens and the MRS predicates for the same parse. The flexibility in defining the semantic contribution of lexical items allowed by HPSG formalism is used extensively in the ERG in the interests of linguistically precise and consistent analysis. Some surface tokens (like copulas and auxiliaries) do not directly contribute predicates in the semantic composition of a sentence. Others contribute several predications (the case of semantic decomposition and generalization). For instance, lexical items *today* and *yesterday* used as verb modifiers contribute four predications each, three of them shared (loosely paraphrased: "at the today time", "at the yesterday time"). This is in contrast to dependency parsers that always work over the original surface tokens.[4]

The distinguished top predication, together with the variables linked across EPs, makes it possible to represent an MRS structure as a graph. However, dealing with an MRS as a flat bag of predications is more computationally tractable, and often sufficient for practical purposes. This is the route I took for the work described here. In the words of Copestake et al:

> "The point of MRS is [...] that it integrates a range of techniques in a way that has proved to be very suitable for large, general-purpose grammars for use in parsing, generation and semantic transfer."(Copestake et al., 2005, 283)

In addition to quantifier scope underspecification, the DELPH-IN grammars and the ERG in particular are underspecified with respect to word senses. The primary grammar design focus is syntax and compositional semantics modeling, and the thus the lexicons generally do not distinguish different word senses for open-class words, as long as these exhibit the same syntactic behavior, and have atomic, non-reducible semantic contribution. This approach ensures that word sense ambiguity is not adding to the computational

[4]For a detailed contrastive study of a variety of syntacto-semantic dependency formats, including MRS representations, see Ivanova et al. (2012).

tractability challenges of parse algorithms. In our example, that means that the *_kid_n_1* is a lexical item that could signify either a child or a young goat. The problem of word sense disambiguation in the DELPH-IN grammars is thus deferred to later processing as it would be a poor candidate for modeling via unification constraints (according to Dan Flickinger, the primary ERG contributor).

It is illuminating to compare the coverage and average ambiguity of open-class words in the ERG and WordNet lexicons, as shown in Table 2.1. The ERG polysemy here was calculated by dividing the number of lexical items by the number of unique lemmas, per part of speech. WordNet lexicon models sense distinctions for open-class words extensively, and in fact is often criticized for drawing too many fine lines between the senses. WordNet is discussed in more detail in Section 2.4. One important thing to note here is that for all parts of speech except adverbs, WordNet has higher average polysemy than the ERG, and since the criteria for sense distinctions in the ERG are quite different from those in WordNet, the mapping between the ERG and WordNet senses is very non-trivial.

| | ERG | | WordNet | | ERG:WordNet |
|---|---|---|---|---|---|
| POS | Total | Avg. polysemy | Total | Avg. polysemy | Polysemy ratio |
| noun | 18,158 | 1.04 | 117,097 | 1.23 | 0.84 |
| verb | 8,229 | 2.03 | 11,488 | 2.16 | 0.94 |
| adjective | 5,603 | 1.08 | 22,141 | 1.41 | 0.76 |
| adverb | 2,055 | 1.30 | 4,601 | 1.24 | 1.05 |

Table 2.1: ERG 1111 and WordNet 3.0 lexicon comparison

### *2.1.1 Dependency MRS (DMRS)*

There exist several MRS representation formats. For my thesis implementation, I work with Dependency MRS, or DMRS, described in Copestake (2009) and similar to the elementary dependencies introduced in Oepen & Lønning (2006). This model represents the semantics of a sentence as a set of labeled elementary predicate (EP) nodes and a set of binary links representing dependencies between the nodes. More specifically, I am working

with the DMRS structures exported into an XML format by the [incr tsdb()] tool (Oepen, 2001), referred to as *DMRX format*.

Figure 2.2 shows an example sentence, *We had lunch with Pauling*, represented in DMRX format. The sentence has five surface words, for which there are eight predicate nodes, 10001 through 10008. Three of the surface tokens (*we*, *lunch* and *Pauling*) correspond to two predicates each: a noun and a quantifier. This is a very common characteristic of the ERG analysis.

Only three of the predicates (10003, 10005 and 10006) are *realpreds*, the rest are *gpreds*. The *realpred* 10005 corresponding to the surface form *lunch* has no arguments: there are no link nodes whose *from* attribute is 10005. The other two *realpreds* (10003 for *had* and 10005 for *with*) take two arguments each, with the bleached names ARG1 ad ARG2. Finally, the *gpreds* which are quantifiers are linked to the other predicates with RSTR-type links (10002 to 10001, 10004 to 10005, 10007 to 10008).

### *2.2 Redwoods Corpora*

The Redwoods Treebank (Oepen et al., 2004) is a dynamic treebank comprising 45,000 (as of the Seventh Growth) sentences, drawing from a variety of domains. Unlike the Penn Tree Bank (Marcus et al., 1993) or Prague Dependency Bank (Hajic, 1998), the Redwoods Treebank is based on an explicit HPSG formulation (specifically, the English Resource Grammar). The automatic ERG analyses are manually disambiguated to select the best analysis. The manual treebanking methodology is based on presenting the annotator with a set of *discriminants*: minimal differences narrowing down the set of possible parses until the correct parse is selected. The discriminant choices are recorded and stored to facilitate the treebank migration to new grammar releases. The dispreferred analyses are also stored to facilitate future updates and for building statistical models. Each analysis is a full HPSG sign that can be exported into a variety of formats.

For this work, I used SemCor and WeScience data sets from Redwoods Treebank Seventh Growth (based on the ERG version 1111). These will be described in more detail in the Methodology chapter.

```
<!--[6031360] (1 of 3) {1} ‘He had lunch with Pauling.’-->
<dmrs cfrom=’-1’ cto=’-1’>
<node nodeid=’10001’ cfrom=’0’ cto=’2’><gpred>pron_rel</gpred>
   <sortinfo cvarsort=’x’ pers=’3’ num=’sg’ gend=’m’ prontype=’std_pron’/>
</node>
<node nodeid=’10002’ cfrom=’0’ cto=’2’><gpred>pronoun_q_rel</gpred><sortinfo/>
</node>
<node nodeid=’10003’ cfrom=’3’ cto=’6’><realpred lemma=’have’ pos=’v’ sense=’1’/>
   <sortinfo cvarsort=’e’ sf=’prop’ tense=’past’ mood=’indicative’ prog=’minus’ perf=’minus’/>
</node>
<node nodeid=’10004’ cfrom=’7’ cto=’12’><gpred>udef_q_rel</gpred><sortinfo/>
</node>
<node nodeid=’10005’ cfrom=’7’ cto=’12’><realpred lemma=’lunch’ pos=’n’ sense=’1’/>
   <sortinfo cvarsort=’x’ pers=’3’ num=’sg’/>
</node>
<node nodeid=’10006’ cfrom=’13’ cto=’17’><realpred lemma=’with’ pos=’p’/>
   <sortinfo cvarsort=’e’ sf=’prop’ tense=’untensed’ mood=’indicative’/>
</node>
<node nodeid=’10007’ cfrom=’18’ cto=’26’><gpred>proper_q_rel</gpred><sortinfo/>
</node>
<node nodeid=’10008’ cfrom=’18’ cto=’26’ carg=’Pauling’><gpred>named_rel</gpred>
   <sortinfo cvarsort=’x’ pers=’3’ num=’sg’ ind=’plus’/>
</node>
<link from=’10002’ to=’10001’><rargname>RSTR</rargname><post>H</post></link>
<link from=’10003’ to=’10001’><rargname>ARG1</rargname><post>NEQ</post></link>
<link from=’10003’ to=’10005’><rargname>ARG2</rargname><post>NEQ</post></link>
<link from=’10004’ to=’10005’><rargname>RSTR</rargname><post>H</post></link>
<link from=’10006’ to=’10003’><rargname>ARG1</rargname><post>EQ</post></link>
<link from=’10006’ to=’10008’><rargname>ARG2</rargname><post>NEQ</post></link>
<link from=’10007’ to=’10008’><rargname>RSTR</rargname><post>H</post></link>
</dmrs>
```

Figure 2.2: Sentence *We had lunch with Pauling* in DMRX format.

### *2.3 HPSG Parse Selection*

Current models for parse selection employed by DELPH-IN HPSG parsers are based on the approach introduced in Toutanova et al. (2002). It is a discriminative MaxEnt ranker using features based on the ERG lexical type names of the leaf nodes, and the tree derivation rules (with up to three-level grandparenting). In some configurations it can also use n-gram smoothing over the surface strings.

The parse selection accuracy based on this approach varies greatly from domain to domain, depending on the source of training data. For SemCor and WeScience data sets, the syntactic model without n-gram smoothing trained on the remainder of the Redwoods Treebank (30,000+ sentences) yielded around 40% parse selection for SemCor and 36% for WeScience. These numbers are realistic for real-life scenarios, where treebanked in-domain data is rarely available.

### *2.4 WordNet and Semantic Files*

Despite the oft-cited complaints about coverage and sense-distinction shortcomings, Princeton WordNet (Fellbaum, 2010) remains the lexical semantic resource of choice for NLP researchers. Wordnets sharing a common lexicographic approach and structure are available for a large and growing number of languages (Bond & Paik, 2012). The Princeton WordNet also serves as a pivot for many other projects providing lexical annotation. Just a few notable examples include Verbnet (Schuler, 2005), the SUMO ontology mapping (Niles & Pease, 2003), OntoNotes (Hovy et al., 2006), and Sentiwordnet (Baccianella et al., 2010). Such resources could be used as an alternative or additional source of semantic categories to enhance the semantic modeling in the future.

One of the shared characteristics of multilingual wordnets are the 45 top-level semantic categories called *semantic files* (SF; also referred to as *lexical files* or *supersenses*). These categories provide a cross-lingual fixed-level lexical semantic abstraction. One thing to note here is that WordNet only provides sense information for open-class words (nouns, verbs, adjectives and adverbs). These generalize the words in sentence (1) as follows: *see* is `verb.perception`, *saw* is `verb.contact` and *kid* is either `noun.person` or `noun.animal`.

Tables 2.2 and 2.3 summarize the lexical files for nouns and verbs, respectively. The examples are drawn from SemCor and include multi-word expressions and named entities in addition to single words.

In addition, the Princeton WordNet structure is informed by corpus observations from SemCor (Miller et al., 1993), the largest sense-annotated data set, based on Brown Corpus (Kučera & Francis, 1967) data. SemCor annotation provided information such as occurrence counts for different word senses. The sense ordering for individual words is based on the SemCor counts, with first senses being the most frequent ones. First senses are frequently used as a (very robust) baseline for word sense disambiguation (WSD) tasks.

I calculated WordNet ambiguity based on SemCor counts cited in the *cntlist* file released with WordNet 3.0. These are not broken down by part of speech. The results are presented in 2.4. Here fine-sense ambiguity refers to the ambiguity with respect to WordNet senses. The coarse-grained ambiguity is ambiguity with respect to semantic files.

### *2.5 MALLET: A Machine Learning for Language Toolkit*

For discriminative modeling, I have used MALLET (McCallum, 2002) – an open source machine learning toolkit, widely used for classification, ranking, sequence labeling, and other machine learning NLP tasks. I used MALLET version 2.0.7 for this work, the most recent at the time of writing, and ran my experiments in the Linux environment.

### *2.6 Summary*

In this chapter, I introduced the conceptual frameworks and tools my thesis is building on. I started by describing HPSG and MRS formalisms. I then briefly reviewed the English Resource Grammar, paying more attention to its lexicon structure. After an overview of the Redwoods Treebank, I described the process for automatic parse ranking employed by the DELPH-IN parsers. I then described WordNet and WordNet semantic files. Finally, I introduced the MALLET toolkit that I am relying on in my experiments.

In the following chapter, I will review recent work aiming to improve parse selection using lexical semantic information, inside and outside DELPH-IN.

| Semantic File | Examples |
|---|---|
| noun.Tops | someone, Venus, Hudson |
| noun.act | dive, labor, drinking |
| noun.animal | dog, chicken, grasshopper |
| noun.artifact | fence, mosaic, telephone_booth |
| noun.attribute | indolence, politeness, obscurity |
| noun.body | arm, brains, stomach |
| noun.cognition | understanding, wishful_thinking, irreverence |
| noun.communication | word, message, warning |
| noun.event | whir, initiation, burial |
| noun.feeling | pleasure, qualms, good_humor |
| noun.food | coffee, radish, pork_chops |
| noun.group | people, small_town, middle_class |
| noun.location | apex, parade_ground, Wyoming |
| noun.motive | urge, reason |
| noun.object | creek, sea, cove |
| noun.person | Watson, Holy_Father, iron_worker |
| noun.phenomenon | fog, rain, blood_pressure |
| noun.plant | willow, tree, crops |
| noun.possession | money, fees, amount |
| noun.process | proliferation, development, unfolding |
| noun.quantity | nothing, pile, half |
| noun.relation | part, connection, West |
| noun.shape | angle, margin, indentations |
| noun.state | priority, vacuum, life |
| noun.substance | marble, charcoal, dust |
| noun.time | summer, November, two_weeks |

Table 2.2: WordNet noun semantic files with examples

| Semantic File | Examples |
|---|---|
| verb.body | laugh, sleep, shave |
| verb.change | come, begin, resettle |
| verb.cognition | think, assume, expect |
| verb.communication | say, scold, reassure |
| verb.competition | win, fight |
| verb.consumption | use, sip, chew |
| verb.contact | hold, lay, tie |
| verb.creation | make, devise, simulate |
| verb.emotion | like, disappoint, attract |
| verb.motion | move, climb, stumble |
| verb.perception | hear, watch, look |
| verb.possession | have, give, lose |
| verb.social | assist, treat, get_married |
| verb.stative | be, stand, regard |
| verb.weather | shine, blow, clear |

Table 2.3: WordNet verb semantic files, with examples

| | |
|---|---|
| Number of ambiguous (unique lemma-pos) items | 24,788 |
| Number of coarse sense disambiguated items | 30,656 |
| Number of fine-sense disambiguated items | 37,198 |
| Percentage of fine-sense ambiguous items | 72.91% |
| Percentage of coarse-sense ambiguous tokens | 50.92% |
| Average fine-sense ambiguity per item | 1.50 |
| Average coarse-sense ambiguity per item | 1.21 |

Table 2.4: WordNet ambiguity based on SemCor counts

# Chapter 3

## LITERATURE SURVEY

In this chapter, I will survey several relatively recent publications that attempted to improve parse selection using semantic information. Some of these attempts were successful, others failed to demonstrate improvements.

Bikel (2000) marked one of the first attempts to integrate word sense information to improve parse ranking, also aiming to simultaneously disambiguate words. The author modified a lexicalized PCFG parser based on BBN SIFT (Miller et al., 2000) and trained it on the intersection of SemCor and Penn Treebank (Marcus et al., 1993). The main extension to the lexicalized features was to include the synsets, using syntactic head relation to its neighboring consituents as a proxy for predicate-argument relations. While the WSD performance reported was good, this method failed to demonstrate any parse selection improvements, presumably because the syntactic relations are not a direct enough reflection of the predicate-argument relations. In addition, synset-level backoff may not be sufficient to overcome sparsity problems, while also failing to capture purely lexicalized patterns.

Work by Fujita et al. (2007, 2010) was a successful attempt to leverage semantic backoff, evaluated in the context of HPSG parse ranking, similar to what I am attempting. The experiments were conducted for Japanese, using the JACY grammar (Siegel & Bender, 2002). The authors used was the Japanese Hinoki Corpus (Bond et al., 2005) of dictionary definitions and example sentences, annotated with both sense information and syntactic preferences, as their data set. A discriminative model was trained using syntactic features, surface-oriented n-gram features, and semantic features defined over elementary predications with the arguments replaced by their semantic categories. They experimented with several levels of semantic categories, following the hypernym chain to determine the best semantic generalization level. The reported results improved on the syntax-only parse selection accuracy by 5.6%, bringing the overall accuracy to 69.4% for dictionary definitions.

My work here is an attempt to build on these results for English, but my approach differs in many respects. While Fujita et al. (2010) used gold sense annotations, I am experimenting with automatic tags (namely, first WordNet senses). This approach means that my results are close to what can be expected in the real-life setting, where gold sense annotations are not available. We use different semantic categories because these experiments were not based on WordNet, but rather on the Goi-Taikei Japanese ontology (Ikehara et al., 1997). I am also evaluating the fixed level backoff approach, which generates far fewer features. Finally, instead of adding lexical semantic features into the mix for building a single discriminative model, I separate them into a model of their own to produce a semantic score. I then combine this semantic score with the original ranker score to re-rank candidate parses.

Two recent works outside of HPSG, Agirre et al. (2008, 2011), provided further evidence for the value of lexical semantic generalization for parsing. Agirre et al. (2008) retrained two statistical parsers, from Bikel (2004) and Charniak (2000), on modified data that included replacing open-class words with their WordNet semantic files (SFs), WordNet synsets and wordform+SF (e.g., *knife+noun.artifact*). The evaluation procedure involved full parsing and PP attachment. Note the first two methods employ semantic generalization, while the third one was a type of semantic specialization. The data set for their experiments was the intersection of Penn Treebank (Marcus et al., 1993) and SemCor. The experiments included gold-sense, first-sense and automatically-tagged data. In addition, the authors had carried out separate experiments for semantic backoff of nouns or verbs only. Their best results yielded error rate reductions of 6.9% for Bikel full parse and 20.5% for Charniak PP attachment, using semantic generalization to semantic files and first sense tags in both cases.

Agirre et al. (2011) followed up on the previous work by re-training the Malt dependency parser (Nivre, 2006). Since their prior experiments indicated that gold sense information was not necessary for parsing improvements, they were able to prepare full Penn Treebank corpus with automatic sense ranking (ASR), following the method from McCarthy et al. (2004). In addition, they tagged words with their WordNet first senses. The resulting experimental setup was very similar to Agirre et al. (2008), with nine configurations: three levels of semantic infomation (SFs, synsets and wordform+SF), and three part

of speech tagging schemes for each (noun-only, verb-only and all-POS). The best labelled attachment score (LAS) absolute improvement of 0.36% (or 2.6% error rate reduction) was obtained with SF annotation over 1st senses for all parts of speech. In addition, unlike the constituency parsers used in their previous work, the Malt parser allowed combining features in the training data, and the authors observed that such combination was helpful, at least on their smaller data subset of SemCor and Penn Treebank intersection.

Drawing on the Agirre et al. (2008, 2011) results, I am using first sense tags for my setup. In addition to being carried out over a unification-based grammar, my approach is different in that I am building a re-ranking model. This means that the syntactic structure and surface-form information can be cleanly separated from the semantic contribution and processed separately.

Most recently, MacKinlay et al. (2012) attempted to integrate lexical semantic classes for the task of HPSG parse selection. Their work was informed by the experiments by Fujita et al. (2010) and Agirre et al. (2008), described earlier in this section. Similar to my set-up, they used a pre-parsed forest of top 500 parses. However, instead of using the original parse ranking scores, they re-created the syntactic features used by the syntactic ranker, added semantic classes and trained a MaxEnt parse selection model using TADM toolkit (Malouf, 2002). The original parse ranker feature set with 3-level grandparenting was recreated and augmented with new features, where word surface forms were replaced with their semantic files (*issues* replaced with `noun.cognition`, for example), and an additional set of features where surface forms were paired with their semantic files. The resulting feature set thus still contained the original lexical information, allowing the discriminative model to learn the weights and use semantic backoff only as necessary. Unlike Fujita et al. (2010), these experiments add the semantic class information to syntactic derivation trees, rather than semantic dependencies. In that respect, the model is somewhat similar to Bikel (2000). The reason the authors chose syntactic features was potential ease of integration with the current parser selective unpacking (Carroll & Oepen, 2005) approach, where the ranking information is needed for efficient processing *before* the semantic structure is produced. These experiments did not yield significant improvements, and hurt in some configurations.

The somewhat mixed results reported in these papers seem to indicate that perhaps lexical semantic information is most useful when its features are closely integrated with structural semantics: the idea I am aiming to evaluate in my experiments. The best results so far have been reported by Fujita et al. (2010) when the features were designed over semantic dependencies. Therefore, in my setup, I am also generating features which model MRS structures.

Another interesting observation from these papers is that the first sense assignment performs at least as well as gold sense tags for semantic backoff. The experiments I conducted use first senses for lexical semantic backoff, thus putting these findings to use (and to test).

# Chapter 4
# METHODOLOGY

In this chapter, I present the methodology used to build the semantic language model, use its output for parse re-ranking and evaluate the results over two Redwoods corpora.

## *4.1 Data*

The data for my experiments was drawn from the Redwoods Corpus (Oepen et al., 2004), introduced in Chapter 2 and distributed as part of the Seventh Growth of the Redwoods Treebank. I used two Redwoods data sets, SemCor and WeScience.

SemCor Redwoods dataset contains sentences from the SemCor corpus (Miller et al., 1993), which, in turn, was taken from the Brown corpus (Kučera & Francis, 1967) and hand-annotated with WordNet word senses. The data selected by Redwoods was tagged for all nouns, verbs, adjectives and adverbs (i.e., "brownv", which tags only verbs, was not included).[1]

The SemCor Redwoods dataset consists of three parts (called *profiles*): SC01, SC02 and SC03. Together, they comprise 2,813 sentences of which 2,493 (or 88.6%) contain exactly one preferred analysis. For my experiments, I used only the 2,493 fully disambiguated items.

Table 4.1 summarizes the parse ambiguity within Redwoods SemCor corpus using the scores from the syntactic MaxEnt parse ranker, trained on all of Redwoods data except SemCor and WeScience, that served as the baseline for re-ranking. The second row represent the baseline system's exact match parse selection accuracy (40.8%).

The WeScience data set (Ytrestøl et al., 2009) (also introduced in Section 2) was used for the second set of experiments. It contains treebanked sentences taken from Wikipedia

[1] A mapping between the individual sentences of the Princeton SemCor corpus and the Redwoods treebank was produced as part of this work. It can be downloaded from `https://sites.google.com/site/zpozen/clms-thesis/SemCoreMapping.csv`.

| # parses | % | Condition |
|---|---|---|
| 68 | 0.027 | only one parse |
| 1018 | 0.408 | parse 1 is preferred |
| 335 | 0.134 | parse 2 is preferred |
| 1517 | 0.608 | preferred parse within top 3 |
| 1895 | 0.760 | preferred parse within top 10 |
| 170 | 0.068 | preferred parse within 10-20 range |

Table 4.1: SemCor original parse selection accuracy (out of 2493 items)

articles about computational linguistics. WeScience data is grouped into 13 profiles, WS01 through WS13: I used all of them for these experiments. Out of the 8,592 sentences in that data set, only 2,553 (or 29.7%) had exactly one selected best parse, underscoring the difficulty of dealing with specialized domain data. The baseline syntactic model over these sentences selected the preferred parse with an accuracy of 36%, and placed the preferred parse within the top 10 with 71.8% accuracy. These measures are again noticeably lower than the corresponding SemCor figures. Table 4.2 summarizes the syntactic ranker results over this WeScience data set portion.

| # parses | % | Condition |
|---|---|---|
| 45 | 0.017 | only one parse |
| 920 | 0.360 | parse 1 is preferred |
| 302 | 0.118 | parse 2 is preferred |
| 1408 | 0.551 | preferred parse within top 3 |
| 1835 | 0.718 | preferred parse within top 10 |
| 211 | 0.082 | preferred parse within 10-20 range |

Table 4.2: WeScience original parse selection accuracy (out of 2553 items)

### *4.2 First Sense Tagging*

Agirre et al. (2008) and Agirre et al. (2011) twice reported somewhat surprising results that showed automatic first sense sense assignments outperforming the hand-annotated gold senses. Though not fully understood at this point, this finding gives hope to the idea of trying automatic first-sense annotation as a foundation for lexical semantic modeling to improve real-life parse selection, when gold sense tags are not available. The experimental setup used in my thesis uses first-sense annotation as a basis for many features. Here, I will describe the steps I took to map Redwoods SemCor and WeScience data sets predicates to their first WordNet senses.

As discussed in Section 2, Princeton WordNet orders word senses based on the SemCor sense-annotated corpus observation. For the purposes of my tagging, first sense is defined as the most frequent fine-grained word sense as observed in the SemCor corpus. The mapping algorithm does not query the frequency information directly, instead relying on the frequency-based sense ranking provided by WordNet itself.

As input to this mapping procedure, I am using MRS sentence representations exported into *DMRX format* and described in Chapter 2.

After experimenting with several more sophisticated strategies, I eventually adopted a very straightforward approach for first sense matching, which seemed to work equally well. WordNet is queried with the lemma and POS tag of each open-class ERG real predicate (specified in the input DMRX), and the first sense returned is used for tagging. Several special cases, described below, are handled differently by my implementation, aiming to maximize the tagging coverage without compromising its accuracy.

If the ERG lemma contains a plus sign character (used as a connector for multiword lemmas), I construct three candidate lemmas to querying WordNet database: one replaces the plus sign with an underscore, another with a space, and the third one removes it altogether. So, for example, *of+course* ERG lemma would produce three alternative lemmas: *of_course*, *of course* and *ofcourse*. All three variants are stored as alternate lemma candidates.

If the ERG lemma contains a dash character (e.g., *ante-bellum*), two additional candidate lemmas are generated: one without the dash (*antebellum*), and one with an underscore in

place of the dash (*ante_bellum*).

If the predicate is a prepositional verb, ERG DMRS usually lists the head verb as the lemma and the preposition as the sense, as shown in (2).

(2) <realpred lemma='give' pos='v' sense='up'/>

I construct a candidate WordNet lemma by connecting these two attributes with an underscore, resulting in *give_up*. This string is used to query the WordNet verb database. The first synset returned is taken to be the first sense. If no synset is found, WordNet is queried with just the head verb lemma (*give*), and, again, the first result is used.[2] If no synsets were returned at this point and there are alternate lemma candidates to try, the process is repeated for each alternate lemma candidate.

For adjectives, WordNet distinguishes head and satellite adjective synset types, with head adjectives expressing the basic qualities and able to form direct antonyms (e.g., *wet, dry*), and satellite adjectives expressing concepts similar in meaning to the head synsets, but not considered directly antonymous (e.g., *sodden, arid*). The ERG makes no such distinction, and therefore, in order to find the first sense for an ERG adjective, I first query WordNet with the adjective lemma for head adjectives, and, if none are returned, re-query for a satellite adjective with the same lemma. The first matching synset returned at this point is taken as the first sense. If neither query returned a synset, I query WordNet for an adverb with the same lemma. This may sound counterintuitive, but many WordNet adverbs are considered adjectives by the ERG (such as *twice, always, never, tomorrow*, etc.). If WordNet returns a match for the adverb query, I take the first result to be the first sense. Otherwise, the process is repeated for each alternate lemma candidate.

If none of the queries described above returned any matches, the predicate is left unmatched and untagged. The most frequent unmatched tokens over SemCor preferred parses are modal verbs, making up 39% of all unmatched tokens. These are absent from WordNet and therefore legitimately excluded. The bulk of the remaining unmatched items are other predicates also absent from WordNet. Table 4.3 shows the top 10 unmatched ERG

[2]This is just a heuristic: the first sense SF for the head verb is of course not necessarily appropriate for the unmatched phrasal verb.

predicates in the SC data set, with their respective token counts.

| Token count | ERG predicate |
|---|---|
| 21 | _should_v_modal_rel |
| 32 | _may_v_modal_rel |
| 34 | _because_x_rel |
| 38 | _must_v_modal_rel |
| 39 | _might_v_modal_rel |
| 50 | _could_v_modal_rel |
| 59 | _if_x_then_rel |
| 81 | _when_x_subord_rel |
| 108 | _in+order+to_x_rel |
| 183 | _would_v_modal_rel |

Table 4.3: Ten most frequent ERG open class realpreds that did not get matched with WordNet first senses, with their SemCor corpus counts.

I output the sense-tagged data in the extended DMRX format, introduced in Chapter 3 and defined in Appendix A, writing out the sense tags as node/sense attributes. The sense tags are optional XML elements and I omit them for predicates for which I cannot find a suitable WordNet sense. Figure 4.1 demonstrates two DMRS nodes decorated with first senses as a result of the process described. Note that for *have*, the correct sense in the expression *have lunch* should be *verb.consumption* rather than *verb.possession*. This highlights the nature of first sense annotation.

Table 4.4 demonstrates the mapping that was produced for the *take* verb family. Table 4.5 summarizes the results of the SemCor and WeScience first sense tagging process. The latter highlights the differences between the two data sets: SemCor has fewer (but more frequent) unmatched types, while WeScience exhibits the opposite trend: there are more unmatched types which occur with less frequency. The SemCor data set, based on the Brown corpus, has many predicates that WordNet considers to be closed class, while for WeScience, predicates are often unmatched due to WordNet's incomplete coverage for specialized domains: for instance, such nouns as *executable*, *networking* or *parse* are not part

of WordNet 3.0.

| ERG predicate | WordNet Semantic File |
|---|---|
| _turn_v_1_rel | verb.motion |
| _turn_v_in_rel | verb.motion |
| _turn_v_on_rel | verb.contact |
| _turn_v_out_rel | verb.stative |
| _turn_v_over_rel | verb.possession |
| _turn_v_prd_rel | verb.motion |
| _turn_v_up_rel | verb.change |

Table 4.4: Mapping of the ERG predicates for *turn* verbs to first WordNet senses

```
<node cto="6" cfrom="3" nodeid="10003">
    <realpred sense="1" pos="v" lemma="have"/>
    <sense offset="2203362" sf="verb.possession" wn_lemma="have"/>
</node>

<node cto="12" cfrom="7" nodeid="10005">
    <realpred sense="1" pos="n" lemma="lunch"/>
    <sense offset="7575076" sf="noun.food" wn_lemma="lunch"/>
</node>
```

Figure 4.1: First sense annotations examples.

### *4.2.1 A note on efficiency*

To speed up the tagging process, the predicates already matched with WordNet first senses are cached in an in-memory hashtable for fast subsequent access. Likewise, predicates that failed to get WordNet matches are cached, so that they can be rejected faster when encountered again.

| | SC preferred | SC dispreferred | WS preferred | WS dispreferred |
|---|---|---|---|---|
| Total types unmatched | 125 | 188 | 184 | 266 |
| Percent types unmatched | 2.2% | 2.8% | 3.3% | 4.0% |
| Total tokens unmatched | 970 | 350,047 | 5,480 | 1,110,558 |
| Percent tokens unmatched | 5.2% | 5.7% | 3.9% | 4.1% |

Table 4.5: Summary of SemCor(SC) and WeScience(WS) first sense mapping results

#### *4.2.2 Alternative first sense mapping approaches*

The first senses could be calculated over semantic files rather than over synsets. It is possible for the most common semantic file, aggregated over several lower count fine-grained senses, to be different from the first synsets semantic file. I have conducted some preliminary experiments calculating the most frequent coarse sense, but this did not seem to improve the overall system performance, while complicating the overall implementation.

In addition, it is possible to use the gold sense mapping described in the previous section to enhance this mapping. The first sense process described so far result in would map all ERG predicates that share a lemma and POS tag to the same WordNet synset. If we were to use the gold sense data, the most frequent sense observed for a specific ERG lexical item could be used instead. For verbs, this could capture some differences in frame semantics.

### *4.3 MaxEnt Ranking*

In this section, I will describe the methodology for creating the MRS ranking model.

#### *4.3.1 MRS Ranking Setup*

After initial rounds of experimentation with generative n-gram models, I followed the example of related work by Fujita et al. (2010) and MacKinlay et al. (2012) and trained a discriminative classifier for the task of MRS ranking. Even though most researchers in the DELPH-IN community use TADM Toolkit for Advanced Discriminative Modeling (Mal-

ouf, 2002), I used the Mallet Language Modeling Toolkit (McCallum, 2002), introduced in the Background section, another powerful package that implements Maximum Entropy classification and has been tested through extensive use in NLP research and applications.

To rank the MRSs corresponding to different parse candidates, I cast the ranking problem as a binary classification between preferred and dispreferred parse MRSs. Each MRS, represented as a set of features based on the predicate relations it describes, becomes a training instance for the MaxEnt (Berger et al., 1996) classifier.[3]

Unlike the generative models, which are trained only on the positive examples (the preferred parses in this case), a discriminative classifier utilizes both positive and negative examples for training. In this case, the negative examples I experimented with varying amounts of negative training data, cutting off the number of top inactive parses at 20, 60, 100, 120 and 200. Figure 4.2 illustrates the learning curve for assigning the highest score to the preferred parse MRS for models, trained on increasing amounts of dispreferred parse data. A a similar trend was observed for SemCor test data and for WeScience. As seen, the performance peaks strongly at around 100 parses, with a sharp dive thereafter. However, there is a possibility that training over the top 500, or more, parses would eventually lead to a precision boost. For this work, however, I am using 100 top parses as the basis for all the reported results.

Instead of adding the MRS features into the syntactic MaxEnt ranker, I am training a separate discriminative classifier over MRS structures. The two-classifier re-ranking approach has been borne out of an intuition that human processing is done separately for syntax and semantics. Humans can parse *The slithy toves did gyre and gimble in the wabe*, as well as recover the meaning of utterances with jumbled syntax and morphology markers (e.g., *Bagel me likes*), as long as they are semantically plausible.

The task is set up as top-N re-ranking. I chose to set N=20 for all the experiments reported here based on the trade-off between oracle accuracy and computational efficiency. Tables 4.1 and 4.2 demonstrate that oracle accuracy is quite high (and hard to approach!)

[3]In this regard, my approach departs from the related DELPH-IN research using TADM in that all my features are optimized globally, rather than per sentence. At decoding time, the comparisons are also made in the global context.

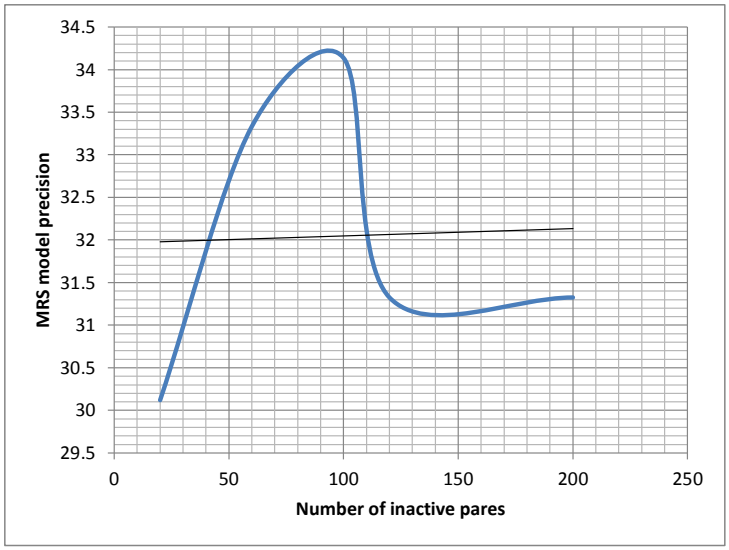


Figure 4.2: Learning curve for MRS ranking precision with POS and lemma features for SemCor dev. The black line represents the linear regression trend.

with N=20. At the same time, the rate of occurrence of correct parses drops very dramatically after the first 20 parse candidates.

During decoding, each parse candidate taken from the top 20 parses (as ranked by the syntactic model) is assigned a score by the MaxEnt classifier representing the likelihood of the parse being active. In fact, in the reported experiments, all the parse candidates from a test (or dev) data set are decoded at once, and therefore the probabilities assigned by the model for the top N candidates for each sentence do *not* add up to 1. This is not a problem for the ranking and re-ranking tasks, however, because I am only interested in how different parses of the same sentence compare.

The MRS ranking model I employed is conceptually similar to the bag-of-words text classification, where each parse candidate can belong to either *active* or *inactive* class, and the "words" are synthetic features representing predicate relations in the MRS. All the features in the MRS representation are thus binary-valued *indicator* features that can be either present or absent in a specific parse.

### *4.3.2 Data split*

Both SemCor and WeScience data sets were split into training, dev and test sets with 0.8 of the items used for training, 0.1 for variable tuning (dev) and 0.1 for test. For SemCor, items with the id range from 6000010 to 6003160, inclusive, went into the test set, 6003170-6006050 became the dev set, and 6006060-6040000 the training set. For WeScience, the ranges are as follows: 10010140-10051970 became test, 10051980-10190460 dev, and 10190470-10860070 training.

| Measure | SC dev | SC test | WS dev | WS test |
|---|---|---|---|---|
| Number of sentences | 249 | 249 | 255 | 255 |
| Number of parses per item (up to 20) | 3,689 | 3,906 | 4,460 | 4,135 |
| Random parse selection baseline | 6.749% | 6.374% | 5.717% | 6.166% |
| Syntactic baseline: correct parse is top 1 | 40.160% | 43.775% | 36.47% | 40.784% |
| Syntactic baseline: correct parse is in top 3 | 60.642% | 64.257% | 52.549% | 63.921% |
| Syntactic baseline: correct parse is in top 10 | 75.903% | 78.313% | 70.196% | 82.745% |

Table 4.6: Dev and test portions of SemCor and WeScience: baseline model for top-20 parses

Table 4.6 shows some interesting measures for the tuning and test portions for both SemCor(SC) and WeScience(WS) data sets. All the re-ranking experiments presented in this thesis are over the top 20 parses, and therefore this data is for the top 20 parses. As can be seen, there is significant variation in the original level of parse ambiguity in the data; and in the baseline model's ability to resolve it. It is interesting to note that the least ambiguous of the 4 sets, SC dev, does not exhibit the highest accuracy for the syntactic model ranking. The most ambiguous set, WS dev, however, does have the lowest syntactic ranking accuracy scores.

#### *4.3.3 Calculating Semantic Scores*

MaxEnt classifier was run with the default Mallet settings. It was fitted with the L-BFGS algorithm (Liu & Nocedal, 1989) with Gaussian prior smoothing.

### ***4.4 Features***

The MRS features used by MaxEnt classifier fall into three broad categories: Predicate-Relation-Predicate (PRP) features, Frame (FR) features and Word-to-Class (W2C) features. In this section, I will describe each of them and give examples.

*Predicate-Relation-Predicate (PRP) Features*

The features in this category represent a relation between two predicates. They take the general form shown in (3).

(3) predicateA_relation_predicateB

The relations draw from the short list of the semantically bleached MRS argument names: ARG1 through ARGn, L-INDEX, R-INDEX, L-HNDL, R-HNDL, ARG and RSTR. The predicates can be either *gpreds* or *realpred* predicates.

A *gpred* predicate is always represented by its name only (e.g., *def_implicit_q_rel*, *pronoun_q_rel*, etc.). The only exception is made for first- and second-person personal pronouns, and for non-neuter third-person singular pronouns. These are additionally represented by *sf:noun.person* — the symbol used for common nouns that belong to *noun.person* semantic file.

A *realpred* can be represented as its lemma, its part of speech, or as its first sense semantic file (sf). For instance, a *realpred* corresponding to the verb *think*, could be represented by *lemma:think*, *pos:v* or *sf:verb.cognition*. I have experimented with using each of these representations separately, and with combining them all. With sf, pos and lemma representations enabled, a DMRS link between the predicate and the first argument for the fragment *she thinks* would generate six PRP features shown in (4).

(4)
- `sf:verb.cognition_ARG1_sf:noun.person`

- `sf:verb.cognition_ARG1_gpred:pron_rel`
- `pos:v_ARG1_sf:noun.person`
- `pos:v_ARG1_gpred:pron_rel`
- `lemma:think_ARG1_sf:noun.person`
- `lemma:think_ARG1_gpred:pron_rel`

Note that a closed-class *realpred* would generally not be mapped to a WordNet sense, and therefore would be represented by its lemma and POS only. For example, the two predicate representations for conjunction *but* would be *lemma:but* and *pos:c*. The same goes for open-class *realpreds* that failed to get mapped.

*Frame (FR) Features*

The predicate relations described so far corresponded to a single link in the DMRS representation. It is desirable to model the way multiple arguments of the same predicate might combine. The setup reported here includes features intended to capture a larger frame segment: a predicate together with its first two arguments. These are represented by the feature template shown in (5).

(5) predicate_arg1_arg2

Both *realpreds* and *gpreds* are modeled using this feature template. Here, *arg1* is the DMRS link target for the DMRS ARG1, L-INDEX or L-HNDL link, and *arg2* is the target node of the ARG2, R-INDEX or R-HNDL link. (6) shows a subset of *realpred* frame features for the sentence fragment *education has priority*, from SemCor item 6000010. Note that I am only displaying matching predicate representations on each end (i.e., only lemma-to-lemma, sf-to-sf, or pos-to-pos), but in the non-ablated setup, all combinations are generated.

(6)
- `pr:lemma:have_arg1:lemma:education_arg2:lemma:priority`
- `pr:pos:v_arg1:pos:n_arg2:pos:n`
- `pr:sf:verb.possession_arg1:sf:noun.act_arg2:sf:noun.state`

To illustrate FR features generated for a *gpred*, I am including (7): another snippet from SemCor 6000010, *girl's education*, featuring the *poss_rel gpred*. Again, only matching *arg1* and *arg2* target representations are included for brevity.

(7)
- `pr:gpred:poss_rel_arg1:lemma:education_arg2:lemma:girl`
- `pr:gpred:poss_rel_arg1:sf:noun.act_arg2:sf:noun.person`
- `pr:gpred:poss_rel_arg1:pos:n_arg2:pos:n`

*Word-to-Class (W2C) Features*

The feature templates described so far represent syntagmatic relations between different predicates, and their various representations. In addition to those, I define unigram features that capture word-to-class assignment likelihoods.

For *realpreds*, I generate the feature template 8 that models lexical ambiguity of the ERG lexicon, reflecting the likelihood of different ERG senses assigned to the same lemma.

(8) `lemma_LEMMA-ERG-sense_erg-sense`

For example, prepositional verbs are a common source of parse ambiguity. Consider parsing a sentence like (9) from SemCor (item 6002220).

(9) At the gate he slowed, looking around.

The ERG can interpret the surface string *looking* as belonging to either *look_v_around* or *look_v_1* lexical items. The first interpretation would generate the feature (10).

(10) `lemma:look_LEMMA-ERG-sense_erg-sense:v_around`

The second interpretation will be represented by (11).

(11) `lemma:look_LEMMA-ERG-sense_erg-sense:v_1`

Finally, a feature template was introduced for *gpreds*, to capture the likelihood of a specific *gpred* posited by the ERG for a specific surface string. It takes the form of feat:surf-gpred.

(12) surf_SURF-GP_gpred

where *surf* takes on the surface sequence in the original sentence, and *gpred* is the *gpred* name. Two examples of feature (12) instantiation are shown in (13). These examples were also chosen to help illustrate the motivation for this feature type, intended to model precisely this type of ambiguity.

(13) `surf:her_SURF-GP_pred:poss_rel`

`surf:her_SURF-GP_pred:pronoun_q_rel`

Note that since many gpreds correspond to whole constituents, feature (12) was coded to only apply to surface strings that contained no spaces, in order to reduce sparsity.

To illustrate how all the features word together to describe a complete MRS, I am including Figure 4.3, showing all the features generated for the preferred parse of sentence (14), drawn from the SemCor corpus (item number 6002140).

(14) But look at us now!

### *4.5 Scoring and Evaluation Procedure*

Coming up with a theoretically sound (or at least plausible) and empirically effective method of combining the scores assigned to parses by the syntactic and semantic classifiers has been one of the challenges of this work. In this section, I will describe the method I employed for all the reported experimental results.

Syntax model scores and the MRS ranker scores are both normalized to z-scores and linearly combined, using the formula given in Equation 4.1.

$$combined\_score = z\_syn\_score - \lambda z\_sem\_score \tag{4.1}$$

*z_syn_score* is the syntactic re-ranker score taken from the baseline system, normalized to Z-scores within top N (where N=20 for all reported experiments). The normalization step makes it possible to compare and combine the two MaxEnt scores, originally computed

```
pos:c_R-HNDL_pos:v
pos:c_R-HNDL_sf:verb.cognition
pos:c_R-HNDL_lemma:look
lemma:but_R-HNDL_pos:v
lemma:but_R-HNDL_sf:verb.cognition
lemma:but_R-HNDL_lemma:look
gpred:pronoun_q_rel_RSTR_gpred:pron_rel
pos:a_ARG1_gpred:time_n_rel
sf:adv.all_ARG1_gpred:time_n_rel
lemma:today_ARG1_gpred:time_n_rel
lemma:today_LEMMA-ERG-sense_erg-sense:a_1
gpred:pronoun_q_rel_RSTR_gpred:pron_rel
gpred:pronoun_q_rel_RSTR_sf:noun.person
surf:us_SURF-GP_pred:pronoun_q_rel
pos:v_ARG2_gpred:pron_rel
pos:v_ARG2_sf:noun.person
sf:verb.cognition_ARG2_gpred:pron_rel
sf:verb.cognition_ARG2_sf:noun.person
lemma:look_ARG2_gpred:pron_rel
lemma:look_ARG2_sf:noun.person
pos:v_ARG1_gpred:pron_rel
sf:verb.cognition_ARG1_gpred:pron_rel
lemma:look_ARG1_gpred:pron_rel
pr:pos:v_arg1:gpred:pron_rel_arg2:gpred:pron_rel
pr:pos:v_arg1:gpred:pron_rel_arg2:sf:noun.person
pr:sf:verb.cognition_arg1:gpred:pron_rel_arg2:gpred:pron_rel
pr:sf:verb.cognition_arg1:gpred:pron_rel_arg2:sf:noun.person
pr:lemma:look_arg1:gpred:pron_rel_arg2:gpred:pron_rel
pr:lemma:look_arg1:gpred:pron_rel_arg2:sf:noun.person
lemma:look_LEMMA-ERG-sense_erg-sense:v_at
gpred:def_implicit_q_rel_RSTR_gpred:time_n_rel
surf:today_SURF-GP_pred:def_implicit_q_rel
gpred:loc_nonsp_rel_ARG1_pos:v
gpred:loc_nonsp_rel_ARG1_sf:verb.cognition
gpred:loc_nonsp_rel_ARG1_lemma:look
gpred:loc_nonsp_rel_ARG2_gpred:time_n_rel
pr:gpred:loc_nonsp_rel_arg1:pos:v_arg2:gpred:time_n_rel
pr:gpred:loc_nonsp_rel_arg1:sf:verb.cognition_arg2:gpred:time_n_rel
pr:gpred:loc_nonsp_rel_arg1:lemma:look_arg2:gpred:time_n_rel
surf:today_SURF-GP_pred:loc_nonsp_rel
```

Figure 4.3: Features generated for the preferred parse of sentence *But look at us now!*

over differently sized candidate pools. The standard Z-score calculation method is used: each raw score is adjusted by the mean and divided by the standard deviation $z = \frac{x-\mu}{\sigma}$. The mean and standard deviation values are calculated over the scores of all candidate parses of the sentence being re-ranked. The resulting scores can be negative or positive.

$\lambda$ is the optimal weight empirically determined as the one giving the best error rate reduction on the dev portion of each data set. A separate optimal weight is determined and applied for top-1, top-3 and top-10 parse selection. It was observed that due to the significant variation between the dev and test sets, the optimal weights on the dev data sets were often far from the optimal on the test data set.

*z_sem_score* is the probability the MRS MaxEnt classifier assigns the the candidate parse being accurate, normalized to Z-scores within top $n$(20) using the same method as with the syntactic scores.

To see score combination in practice, let's look again at (14). Its preferred parse was originally the second, but the semantic model was able to correct this. Table 4.7 demonstrates the score re-ranking with real score numbers for the top three parse candidates (with $\lambda = 0.35$).

| Correct reading? | Syntactic score | Semantic score | Combined Score |
|---|---|---|---|
| no | 2.307 | 0.321 | 2.420 |
| yes | 1.898 | 3.536 | 3.136 |
| no | 0.981 | -0.282 | 0.882 |

Table 4.7: Score adjustment for the example sentence (14)

The re-ranker evaluation procedure is *exact parse match*, i.e., we count the selection as accurate if the combined score for the preferred parse is the maximal of all candidate combined scores. In cases where several MRS parse representations get the same score (either because the two parse candidates have the same semantics, or because the features I used for MRS ranking do not reflect the distinction between the parses), the selection is still counted as accurate if the preferred parse gets the highest MRS score. This may be too

generous at this point for MRS parse selection, since even my fullest set of features still doesn't include some interesting MRS distinctions (verb tense and aspect, for example). The problem does not exist for re-ranking, however, because the syntactic model assigns unique scores to each parse candidate, so that when these are added to the same semantic scores, the resulting combined scores become distinct.

### *4.6 Summary*

In this chapter, I presented the methodology used to set up my experiments. The following chapter will present the experimental results obtained using these procedures.

# Chapter 5

# RESULTS AND ERROR ANALYSIS

This chapter presents the experimental results. I include the results for MRS-based parse selection and combined model re-ranking for several feature sets for both SemCor and WeScience. I also report on the experiments where SemCor-trained model was used for WeScience, and vice versa, to simulate out-of-domain training scenarios. Finally, I take a closer look at the remaining errors in search for future improvement directions.

## *5.1 Results - SemCor*

Table 5.1 presents the baseline measures for SemCor top-20 ranking and re-ranking, for dev and test sets. *Syntactic Model (top N)* rows show the top-N selection precision for the baseline syntax system. I am reporting top-1, top-3 and top-10 numbers, since these are the settings for which I conducted all of my re-ranking experiments. *Oracle (top 20)* is the upper limit on the selection accuracy that can be obtained for this setup: the percent of preferred parses in the top 20. *Random parse baseline* row represents the lower limit. It is higher than 5% (1/20) because some sentences have fewer than 20 parses.

This table highlights the considerable data variation between the dev and test sets. The syntactic ranker appears to have an easier time correctly classifying the test set parses, despite the fact that the random parse baseline is lower (i.e., the test sentences are somewhat more ambiguous). This difference does not invalidate the re-ranking results, and in fact likely makes the parameter tuning perform worse than it could if the two sets were behaving more uniformly.

Table 5.2 presents the results for MRS model-based ranking and re-ranking over the SemCor data set. The table is broken into five sections, depending on the *realpred* predicate representation features used. *No realpreds* is the smallest model, where only features involving gpreds are generated (i.e., any features involving realpreds are skipped alto-

| Precision | dev | test |
|---|---|---|
| Oracle (top 20) | 81.93 | 86.75 |
| Syntactic Model (top 1) | 40.16 | 43.78 |
| Syntactic Model (top 3) | 60.64 | 64.26 |
| Syntactic Model (top 10) | 75.90 | 78.31 |
| Random parse (top1) | 6.75 | 6.37 |

Table 5.1: SemCor baseline measures

gether). *Lemma* row shows results for the feature set where, in addition to gpreds, any realpreds predicates are represented as lemmas. Similarly, *SF* configuration represents realpreds as their first sense semantic files, and *POS* — as their POS tags. The final portion of the table shows the results for a feature configuration where all three predicate representations (lemmas, SF and POS) were used in all possible combinations. All *Top N* re-ranking results are reported as both absolute accuracy and as relative error reduction compared to the syntactic model (ER).

In addition to the top N re-ranking results, I report MRS model accuracy and random MRS baseline for each feature set. Neither one of these measures involve variable tuning, meaning that the results for dev and test sets are equally interesting, and the differences are reflective of the data variation.

Random MRS baseline represents the probability of selecting the correct parse candidate when picking a unique MRS score at random. This baseline varies between MRS feature sets considerably, depending on their ability to express differences between the parses. In cases where the differences are not modeled through the feature set, several parse candidates receive the same score from the MRS classifier. It is therefore necessary to compare the random MRS baseline with the random parse baseline measure from Table 5.1.

MRS model accuracy represents the probability that the highest-scoring MRS corresponds to the preferred parse. This measure needs to be interpreted in conjunction with the random MRS baseline: the high test score for the *No realpreds* feature set is not really

| | dev | | test | |
|---|---|---|---|---|
| | Accuracy | ER | Accuracy | ER |
| No realpreds | | | | |
| Top 1 | 42.17 | 3.36 | 45.38 | 2.86 |
| Top 3 | 62.65 | 5.10 | 63.86 | −1.12 |
| Top 10 | 76.71 | 3.33 | 78.31 | 0.00 |
| Random MRS baseline | 13.06 | | 12.63 | |
| MRS model accuracy | 30.52 | | 34.54 | |
| Lemma | | | | |
| Top 1 | 43.37 | 5.37 | 42.17 | −2.86 |
| Top 3 | 65.06 | 11.22 | 61.85 | −6.74 |
| Top 10 | 77.91 | 8.33 | 78.31 | 0.00 |
| Random MRS baseline | 10.18 | | 9.62 | |
| MRS model accuracy | 30.92 | | 32.93 | |
| SF | | | | |
| Top 1 | 40.16 | 0.00 | 43.37 | −0.71 |
| Top 3 | 62.25 | 4.08 | 65.46 | 3.37 |
| Top 10 | 77.51 | 6.67 | 79.92 | 7.41 |
| Random MRS baseline | 8.68 | | 8.31 | |
| MRS model accuracy | 34.14 | | 28.92 | |
| POS | | | | |
| Top 1 | 40.96 | 1.34 | 43.37 | −0.71 |
| Top 3 | 62.25 | 4.08 | 64.66 | 1.12 |
| Top 10 | 77.11 | 5.00 | 78.31 | 0.00 |
| Random MRS baseline | 9.15 | | 8.84 | |
| MRS model accuracy | 34.14 | | 30.92 | |
| Lemma+SF+POS | | | | |
| Top 1 | 42.17 | 3.36 | 40.96 | −5.00 |
| Top 3 | 63.45 | 7.14 | 64.66 | 1.12 |
| Top 10 | 77.91 | 8.33 | 79.52 | 5.56 |
| Random MRS baseline | 8.39 | | 8.08 | |
| MRS model accuracy | 34.14 | | 34.94 | |

Table 5.2: SemCor results for top-100 trained models

meaningful against the high random MRS baseline, where many parses received the same MRS classifier score. In general, this score is most telling when the random MRS baseline approaches the random parse baseline (like it is in the case of the *Lemma+SF+POS* feature set).

The following observations can be made about Table 5.2 results:

- Lowest random MRS baseline scores are achieved with the *Lemma+SF+POS* features set. This is expected, since more features make it possible to better distinguish between different parses.

- Highest MRS accuracy is achieved with all features enabled, for both dev and test. Highest accuracy of 34.939% is achieved on test — far exceeding the random baseline of 8.08%. However, it is still far from the syntactic model baseline of 43.775%.

- It is not very clear which of the 3 realpred representations is the most beneficial: Lemmas perform strongly for re-ranking on dev, but SFs are the best on test, and POS performance appears reasonable as well, especially for MRS selection accuracy.

- All MRS model accuracy scores are well above the random baselines for their respective feature set.

- For dev data set, all *Top N* re-ranking results are non-negative, with the biggest improvements observed for *Lemma* representations: 5.37% relative error reduction for top-1, 8.33% for top 10, and an impressive 11.22% relative (or 4.42% absolute) error reduction for top-3 re-ranking.

- The re-ranking results for test are mixed and only stayed non-negative for top-10 re-ranking. The best top-10 re-ranking result of 7.41% relative error reduction on test is seen with SF features.

Table 5.3 follows the same structure as Table 5.2 and presents the results of the same experiments, but with the MRS model trained on WeScience data. This is intended to measure the effectiveness of proposed features when in-domain training data is not available.

| | dev | | test | |
|---|---|---|---|---|
| | Accuracy | ER | Accuracy | ER |
| No realpreds | | | | |
| Top 1 | 40.16 | 0.00 | 43.37 | −0.71 |
| Top 3 | 62.65 | 5.10 | 59.84 | −12.36 |
| Top 10 | 77.11 | 5.00 | 78.31 | 0.00 |
| Random MRS baseline | 14.22 | | 13.99 | |
| MRS model accuracy | 24.50 | | 25.70 | |
| Lemma | | | | |
| Top 1 | 40.16 | 0.00 | 43.37 | −0.71 |
| Top 3 | 62.65 | 5.10 | 62.25 | −5.62 |
| Top 10 | 77.11 | 5.00 | 80.32 | 9.26 |
| Random MRS baseline | 11.36 | | 10.77 | |
| MRS model accuracy | 24.90 | | 28.92 | |
| SF | | | | |
| Top 1 | 40.16 | 0.00 | 43.37 | −0.71 |
| Top 3 | 61.04 | 1.02 | 63.86 | −1.12 |
| Top 10 | 77.91 | 8.33 | 78.72 | 1.85 |
| Random MRS baseline | 8.86 | | 8.58 | |
| MRS model accuracy | 21.29 | | 22.49 | |
| POS | | | | |
| Top 1 | 40.16 | 0.00 | 43.37 | −0.71 |
| Top 3 | 61.04 | 1.02 | 63.45 | −2.25 |
| Top 10 | 75.90 | 0.00 | 78.72 | 1.85 |
| Random MRS baseline | 9.18 | | 9.06 | |
| MRS model accuracy | 23.69 | | 27.71 | |
| Lemma+SF+POS | | | | |
| Top 1 | 40.16 | 0.00 | 43.37 | −0.71 |
| Top 3 | 61.04 | 1.02 | 64.26 | 0.00 |
| Top 10 | 76.31 | 1.67 | 79.12 | 3.70 |
| Random MRS baseline | 8.44 | | 8.17 | |
| MRS model accuracy | 25.70 | | 24.50 | |

Table 5.3: SemCor results with WeScience-trained models

I am also reporting results for isolated feature sets, in order to observe how different predicate representations hold across domains. This data shows the following:

- As expected, with out-of-domain training data, MRS selection drops significantly (almost 13 points for SF-only on dev), but still stays well above the random baseline.

- Top-10 re-ranking is still benefiting from the re-ranked scores for both dev and test; top-3 is improved only on dev.

- Top-1 results are not improved for any of the feature combinations, even on dev.

- *Lemma* realpred features are still performing well for both dev and test for MRS selection, despite significant vocabulary differences between SemCor and WeScience domains. The best top-10 parse re-ranking error reduction of 9.26% is achieved with Lemmas on test.

- SF features perform reasonably well for top 10 re-ranking. POS seems to work worse for re-ranking, but yields better MRS model accuracy. One possible explanation is that POS data, unlike SFs, is not sufficiently different from the syntactic model features, and therefore does not add enough new information to re-rank effectively.

### *5.2 Results - WeScience*

Table 5.4 shows the baseline measures for WeScience corpus, and Tables 5.5 and 5.6 summarize the re-ranking results for that data set. The same experiments were conducted for WeScience as for SemCor, and the results are presented in the same format as the SemCor results.

Table 5.4 demonstrates that there is an even more pronounced difference between dev and test data on WeScience than there was for SemCor, suggesting that a stratified cross-validation would be a better approach for this data set.[1] The test data set is both less ambiguous and much better handled by the syntactic model, with the top-1 selection of 40.78% (vs. dev 36.47%).

| Precision | dev | test |
|---|---|---|
| Oracle | 78.43 | 87.84 |
| Syntactic Model (top 1) | 36.47 | 40.78 |
| Syntactic Model (top 3) | 52.55 | 63.92 |
| Syntactic Model (top 10) | 70.20 | 82.75 |
| Random parse (top1) | 5.72 | 6.17 |

Table 5.4: WeScience baseline measures

The experimental results presented in Table 5.5 are generally consistent with the SemCor results with in-domain training data shown in 5.2, but appear to be quite a bit stronger. With all features enabled, the results for all re-ranking brackets are positive for both dev and test. Top-1 selection achieved error reduction of 3.09% (or 1.96% absolute) on dev, and 1.99% (1.18% absolute) on test. Another exciting number is the 40.39% MRS selection accuracy on test with all features enabled, which comes very close to the 40.78% syntactic accuracy. It is especially encouraging when one considers that the syntactic baseline

[1] In fact, the WeScience data is contiguous: the sentences appear in the order they appeared in the original Wikipedia articles. This makes re-shuffling or a stratified data selection even more important for a more reliable experimental setup.

had been trained on much more data (all of the Redwoods Treebank, minus SemCor and WeScience) than the MRS model.

Lemma-based realpred features seem to be performing even better for WeScience than they do for SemCor, with the gains holding for test, as well. This might be due to the more formulaic language within the specialized domain, as compared to the SemCor data, largely drawn from literary fiction.

The fact that SF representation performs well for WeScience data set is noteworthy. First senses for words are calculated based on the observed SemCor frequency, and, given the significant domain difference between WeScience and SemCor data sets, were not necessarily expected to perform well for WeScience.

Finally, Table 5.6 presents experimental results for WeScience data classified and re-ranked using SemCor-trained MRS model. Once again, the MRS selection accuracy takes a significant plunge from the in-domain numbers, but remains well above the random baselines. All features combined provide the strongest MRS selection accuracy performance on both dev and test.[2] The lemma feature set appears to be the best when taken in isolation, followed by POS. In fact, the re-ranking results for both Lemma and POS are stronger than those of the combined feature set, for both dev and test. This last observation is not easy to interpret, but suggests that an ensemble technique might be more effective instead of adding all the features into the same classifier.

[2]The dev result is strongest when normalized by the random baseline.

| | dev | | test | |
|---|---|---|---|---|
| | Accuracy | ER | Accuracy | ER |
| No realpreds | | | | |
| Top 1 | 36.47 | 0.00 | 41.18 | 0.66 |
| Top 3 | 55.29 | 5.79 | 65.88 | 5.44 |
| Top 10 | 73.33 | 10.53 | 82.75 | 0.00 |
| Random MRS baseline | 12.43 | | 12.02 | |
| MRS model accuracy | 22.35 | | 30.59 | |
| Lemma | | | | |
| Top 1 | 36.47 | 0.00 | 40.78 | 0.00 |
| Top 3 | 54.51 | 4.13 | 65.49 | 4.35 |
| Top 10 | 74.12 | 13.16 | 84.71 | 11.36 |
| Random MRS baseline | 8.39 | | 8.82 | |
| MRS model accuracy | 26.27 | | 30.98 | |
| SF | | | | |
| Top 1 | 37.26 | 1.24 | 40.00 | −1.33 |
| Top 3 | 56.47 | 8.26 | 64.31 | 1.09 |
| Top 10 | 72.16 | 6.58 | 81.57 | −6.82 |
| Random MRS baseline | 7.16 | | 7.91 | |
| MRS model accuracy | 23.14 | | 33.73 | |
| POS | | | | |
| Top 1 | 36.86 | 0.62 | 41.18 | 0.66 |
| Top 3 | 57.26 | 9.92 | 65.10 | 3.26 |
| Top 10 | 73.73 | 11.84 | 83.92 | 6.82 |
| Random MRS baseline | 7.92 | | 8.53 | |
| MRS model accuracy | 27.45 | | 34.90 | |
| Lemma+SF+POS | | | | |
| Top 1 | 38.43 | 3.09 | 41.96 | 1.99 |
| Top 3 | 54.90 | 4.96 | 64.31 | 1.09 |
| Top 10 | 71.77 | 5.26 | 83.92 | 6.82 |
| Random MRS baseline | 6.88 | | 7.67 | |
| MRS model accuracy | 27.45 | | 40.39 | |

Table 5.5: WeScience results for top-100 trained models

| | dev | | test | |
|---|---|---|---|---|
| | Accuracy | ER | Accuracy | ER |
| No realpreds | | | | |
| Top 1 | 36.47 | 0.00 | 40.78 | 0.00 |
| Top 3 | 53.33 | 1.65 | 63.92 | 0.00 |
| Top 10 | 72.55 | 7.90 | 83.14 | 2.27 |
| Random MRS baseline | 14.26 | | 14.16 | |
| MRS model accuracy | 25.49 | | 29.02 | |
| Lemma | | | | |
| Top 1 | 36.47 | 0.00 | 40.78 | 0.00 |
| Top 3 | 54.51 | 4.13 | 65.10 | 3.26 |
| Top 10 | 73.73 | 11.84 | 83.14 | 2.27 |
| Random MRS baseline | 10.43 | | 10.90 | |
| MRS model accuracy | 20.00 | | 30.20 | |
| SF | | | | |
| Top 1 | 36.47 | 0.00 | 40.78 | 0.00 |
| Top 3 | 55.69 | 6.61 | 63.14 | −2.17 |
| Top 10 | 70.98 | 2.63 | 82.75 | 0.00 |
| Random MRS baseline | 7.21 | | 8.13 | |
| MRS model accuracy | 20.39 | | 27.84 | |
| POS | | | | |
| Top 1 | 36.47 | 0.00 | 40.78 | 0.00 |
| Top 3 | 56.08 | 7.44 | 62.75 | −3.26 |
| Top 10 | 71.77 | 5.26 | 82.75 | 0.00 |
| Random MRS baseline | 8.18 | | 8.79 | |
| MRS model accuracy | 23.14 | | 30.20 | |
| Lemma+SF+POS | | | | |
| Top 1 | 36.86 | 0.62 | 41.57 | 1.33 |
| Top 3 | 55.69 | 6.61 | 63.14 | −2.17 |
| Top 10 | 70.59 | 1.32 | 82.75 | 0.00 |
| Random MRS baseline | 6.98 | | 7.77 | |
| MRS model accuracy | 22.75 | | 30.59 | |

Table 5.6: WeScience results with SemCor-trained models

### *5.3 Error Analysis*

In this section, I will take a closer look at the areas where the proposed methodology is falling short. I selected SemCor test data (249 sentences), classified using SemCor training data, for this analysis. To recap the from Table and 5.1 Table 5.2, the baseline syntactic model had the accuracy of 43.78%, and the best MRS configuration, Lemma+SF+POS, gave the highest score to the preferred parse in 34.94% of cases, or to 109 items. The semantic model assigned the highest score to 87 of the preferred parses, 25 of which were different from what the syntactic model's selections. This indicates that the two models have significantly different, non-correlated preference criteria, and validates the overall approach and the current set of features.

Sometimes, the DMRS representations of two parses are identical, or the differences are small and not modeled through the features proposed here. To get a better understanding of this phenomenon, I looked at the 9 sentences where the preferred parse was the second-highest ranking according to the syntactic model. Out of those, 5 had the same DMRS representation for the top two parses, meaning that the semantic score could not have been different. In the remaining 4 cases, the differences were in the DMRS *sortinfo* attribute values, not currently modeled. These included gender (*neuter* vs. underspecified) and mood (*indicative* vs. *subjunctive*). Adding features to reflect some or all the *sortinfo* attributes would be straightforward and could help improve the scores.

Perhaps the weakest point of the presented methodology is the score combination (see Equation 4.1 on page 34). The semantic score $\lambda$ weights, as selected based on the highest observed error reduction (ER) numbers for the SemCor dev data set are 1.55 for top 1, 1.35 for top 3 and 3.50 for top 10. These weights were not optimal for the test data set. To illustrate the variable tuning challenge, I plotted error reduction as a function of semantic score weight for SemCor dev and test data in Figures 5.1, 5.2 and 5.3, with weights up to 15.0 attempted. For all three parse selection brackets, the test data set appears to be benefiting far less than the dev set from the score combination. The top 1 selection has a single spike when the error reduction is 0.71% with the semantic model weight of 0.45, and quickly descends into the negative territory. The error reduction for the top 3 bracket is a

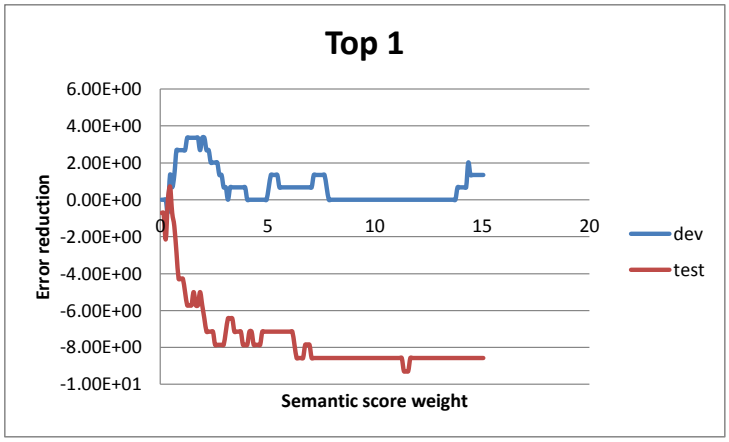


Figure 5.1: Top-1 error reduction with combined scores as a function of semantic score weight for SemCor dev and test data.

lot more stable across weights, and stays positive between 0.25 and 2.0, peaking with the error reduction of 3.37% with the $\lambda$ weight of 1.35. The function shape is also remarkably similar to that of the dev set, but with lower overall scores. Finally, top 10 results are the most stable. In addition to staying positive for all weights for both dev and test, they do not fluctuate much, making them easy to tune.

In summary, the error analysis indicates that the adopted approach is valid and could result in bigger gains with a better re-ranking methodology.

### *5.4 Summary*

I presented the experimental results for parse selection based on MRS features and top-20 parse re-ranking using a combination of baseline syntactic ranker scores and MRS MaxEnt ranker scores. The experiments were conducted over SemCor and WeScience data sets, with both in-domain and out-of-domain training data. I also reported on results of several feature sets, seeking to compare and combine several alternate predicate representations.

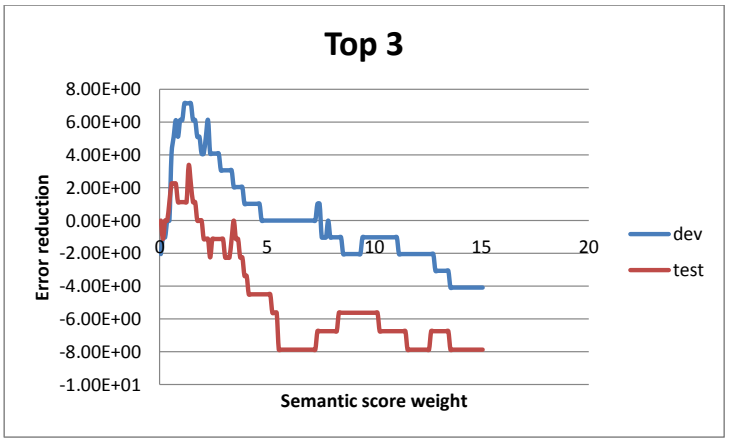


Figure 5.2: Top-3 error reduction with combined scores as a function of semantic score weight for SemCor dev and test data.

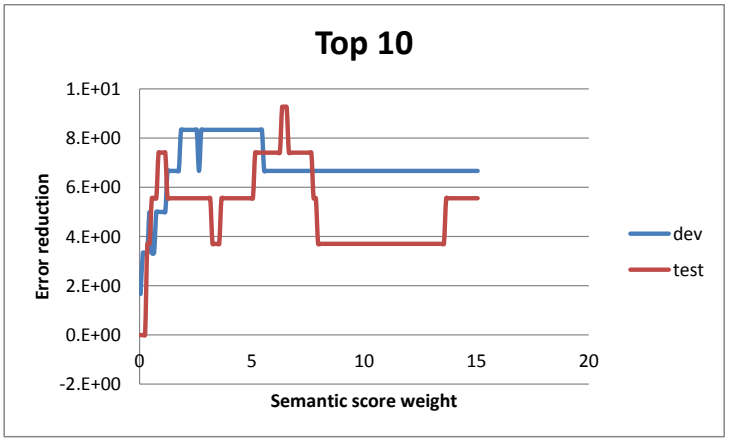


Figure 5.3: Top-10 error reduction with combined scores as a function of semantic score weight for SemCor dev and test data.

# Chapter 6

# CONCLUSIONS AND FUTURE WORK

In this thesis, I explored the idea that monostratal grammars and parsers (like DELPH-IN HPSG parsers) that output semantic representations present an additional opportunity to improve parse selection, by exploiting structural and lexical semantics regularities.

I have presented a series of experiments for HPSG parse selection based on the MRS semantic representations of parse candidates. The method involves modeling predicate-to-predicate link relations intended to capture the likelihoods of different predicate-argument co-occurrences, as well as predicate assignment probabilities for various surface forms. The resulting features were used for training a MaxEnt classifier over top N parse candidates produced by the baseline syntactic parse selection model. I experimented with several ways of representing the predicates: using lemmas, POS tags, and backed-off semantic categories (WordNet semantic files for their first senses). I have attempted and reported each of these predicate representations separately and in combination, and reported on the results. The best model used all three representations, and achieved the MRS selection accuracy of 40.40% for WeScience, getting very close to the syntactic ranker accuracy of 40.78%, despite being trained on much less data.

In addition, in the course of this thesis work, I have developed two methods for tagging MRS representations with word sense information. The gold-sense tagging method for projecting the SemCor (Miller et al., 1993) word sense annotations onto the ERG DMRS representations was applied to the 2,590 sentences of the Redwoods SemCor data set, producing the largest sense-annotated ERG corpus to date. I have not used this data for the present research, but may do so in the future, and sincerely hope other researchers interested in the intersection of lexical and compositional semantics might find it useful. In addition, if a larger subset of SemCor sentences becomes available in the Redwoords Treebank in the future, the automatic sense-tagging process I developed could be applied to

those items as well, to add the word sense annotations.

The first-sense tagging method is also fully automated and does not rely on the availability of hand-annotated data (like SemCor). The presented results indicate that this method of tagging is also effective for MRS selection, and could be easily applied to any data set.

### *6.1 Future Work*

The experiments presented here are by no means exhaustive and in fact are more of a proof of concept. The numbers reported are likely far from the upper bound of what could be achieved on the basis of the general methodology explored here. Many areas for further exploration are possible; I will outline some of them here.

The most obvious weakness of the presented methodology is its very naïve, and not too effective, method for combining original MaxEnt ranker scores with the semantic scores. Other score normalization and weight tuning techniques could potentially yield better results based on the same original scores. Another simple approach to try would be a classification task that includes the two classifier scores as features. Finally, adding the MRS features into the syntactic ranker would side-step the need for score combination altogether.

While the strong performance of MRS-based parse selection model is encouraging, the feature set can be further improved. As mentioned in the error analysis, many MRS elements are not modeled through the features, and some of them (like *mood* and *gender*) often distinguish preferred parses. More detailed feature modeling could be helpful. I have also observed in the course of early experiments over gold-sense tagged data that adding named entity recognition and backing off named entities to the same semantic categories as the common nouns (such as *noun.person* and *noun.location*) provided a boost to the scores. Incorporating a robust NER pre-processing step for automatic sense tagging could provide another score improvement. With the more detailed and therefore more sparse features, it will be important to extend the training data set. Luckily, the sense-tagging methods are fully automated and could be applied to the tens of thousands of disambiguated parses

available through the LinGO Redwoods Treebank.

The first-sense tagging scheme could also be modified based on the observations from the gold-sense tagged data (as described in Appendix B). The observations from the gold-tagging could lead to different first-sense assignments for the ERG predicates that map to the same lemma/POS and currently get the same first sense tags.[1] Also, instead of using first senses, we could integrate an automatic WSD component and use its output for sense tagging (this has, however, been tried by Agirre et al. (2011) and not shown to help, at least in their setup).

For semantic backoff features, I only experimented with coarse semantic categorization at the level of WordNet semantic files here, but I believe a similar approach could be extended to include other types of backoff categories or compositional markers. Other lexical semantic resources mapped to WordNet could be explored for that purpose. For instance, for nouns, CoreLex (Buitelaar, 1998) categories could be a good alternative. The CoreLex backoff is interesting in that, like first-sense assignment, it avoids the need for word sense disambiguation, exploring the nature of systematic polysemy instead, where ambiguous nous are represented as a combination of *all* of their main senses. For verbs, Verbnet (Schuler, 2005) categories could be explored, drawing on syntactic behavior similarities between the verbs.[2] Other lexical resources mapped to WordNet include the SUMO ontology mapping (Niles & Pease, 2003), OntoNotes (Hovy et al., 2006), and Sentiwordnet (Baccianella et al., 2010). In general, the idea of being able to evaluate a variety of lexical semantic models computationally and within a practical task is very intriguing. These could range from componential analysis (Katz & Fodor, 1963; Jackendoff, 1992) to generative lexicon (Pustejovsky, 1991) to semantic primitives (Wierzbicka, 1996), semantic categorization and cognitively-biased ontologies (Buitelaar, 1998; Gangemi et al., 2002), and more.

Finally, it would be most interesting to conduct a cross-lingual experiment, where the semantic model trained on one language is tested on another. JACY (Siegel & Bender, 2002) is the Japanese language HPSG grammar that has broad coverage comparable to the ERG

[1] It would be important to get more of the SemCor sentences treebanked in addition to the current 2590 in order to get more reliable frequency distributions.

[2] Obviously, this would require developing a mapping from VerbNet to the ERG lexicon.

and a similar set of *gpreds*. Semantic file backoff would harmonize the open-class lexical items across the two languages. This would involve the task of mapping closed-class real predicates between English and Japanese.

## BIBLIOGRAPHY

Agirre, E., T. Baldwin & D. Martinez. 2008. Improving parsing and PP attachment performance with sense information. *Proceedings of ACL-08: HLT* 317–325.

Agirre, E., K. Bengoetxea, K. Gojenola & J. Nivre. 2011. Improving dependency parsing with semantic classes. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics* 699–703.

Baccianella, S., A. Esuli & F. Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the seventh conference on international language resources and evaluation (lrec10), valletta, malta, may. european language resources association (elra)*, .

Bender, E.M., D. Flickinger & S. Oepen. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the 2002 workshop on grammar engineering and evaluation-volume 15*, 1–7. Association for Computational Linguistics.

Berger, Adam L, Vincent J Della Pietra & Stephen A Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational linguistics* 22(1). 39–71.

Bikel, Daniel M. 2000. A statistical model for parsing and word-sense disambiguation. In *Proceedings of the joint sigdat conference on empirical methods in natural language processing and very large corpora*, 155–163. Hong Kong.

Bikel, D.M. 2004. Intricacies of Collins' parsing model. *Computational Linguistics* 30(4). 479–511.

Bond, F., S. Fujita, C. Hashimoto, K. Kasahara, S. Nariyama, E. Nichols, A. Ohtani, T. Tanaka & S. Amano. 2005. The Hinoki treebank — a treebank for text understanding. *Natural Language Processing–IJCNLP 2004* 158–167.

Bond, Francis & Kyonghee Paik. 2012. A survey of wordnets and their licenses. In *Proceedings of the 6th global wordnet conference (gwc 2012)*, Matsue. 64–71.

Buitelaar, P. 1998. *Corelex: systematic polysemy and underspecification*: Citeseer dissertation.

Callmeier, U. 2000. PET—a platform for experimentation with efficient HPSG processing techniques. *Natural Language Engineering* 6(1). 99–107.

Carpenter, B. & G. Penn. 1994. ALE: the attribute logic engine user's guide, version 2.0. 1 .

Carroll, J. & S. Oepen. 2005. High efficiency realization for a wide-coverage unification grammar. *Natural Language Processing–IJCNLP 2005* 165–176.

Charniak, E. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st north american chapter of the association for computational linguistics conference*, 132–139. Morgan Kaufmann Publishers Inc.

Copestake, A. 2002. *Implementing typed feature structure grammars*, vol. 110. CSLI publications Stanford, CA.

Copestake, A. 2009. Slacker semantics: Why superficiality, dependency and avoidance of commitment can be the right way to go. In *Proceedings of the 12th conference of the european chapter of the association for computational linguistics*, 1–9. Association for Computational Linguistics.

Copestake, A., D. Flickinger, C. Pollard & I.A. Sag. 2005. Minimal recursion semantics: An introduction. *Research on Language & Computation* 3(2). 281–332.

Edmonds, P. & S. Cotton. 2001. Senseval-2: Overview. In *Proceedings of*, 1–6.

Fellbaum, C. 2010. Wordnet. *Theory and Applications of Ontology: Computer Applications* 231–243.

Fellbaum, Christine (ed.). 1998. *WordNet: An electronic lexical database*. MIT Press.

Flickinger, Dan. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering* 6 (1). 15 – 28.

Fujita, Sanae, Francis Bond, Stephan Oepen & Takaaki Tanaka. 2007. Exploiting semantic information for HPSG parse selection. In *Acl 2007 workshop on deep linguistic processing*, 25–32. Prague, Czech Republic: Association for Computational Linguistics. `pubs/ACL2007_DLP04.pdf`.

Fujita, Sanae, Francis Bond, Takaaki Tanaka & Stephan Oepen. 2010. Exploiting semantic information for HPSG parse selection. *Research on Language and Computation* 8(1). 1–22.

Gangemi, A., N. Guarino, C. Masolo, A. Oltramari & L. Schneider. 2002. Sweetening ontologies with DOLCE. *Knowledge engineering and knowledge management: Ontologies and the semantic Web* 223–233.

Hajic, J. 1998. Building a syntactically annotated corpus: The Prague dependency treebank. *Issues of valency and meaning* 106–132.

Hovy, E., M. Marcus, M. Palmer, L. Ramshaw & R. Weischedel. 2006. OntoNotes: the 90% solution. In *Proceedings of the human language technology conference of the naacl, companion volume: Short papers*, 57–60. Association for Computational Linguistics.

Ikehara, S., M. Miyazaki, S. Shirai, A. Yokoo, H. Nakaiwa, K. Ogura, Y. Ooyama & Y. Hayashi. 1997. Goi-Taikei — a Japanese lexicon.

Ivanova, A., S. Oepen, L. Øvrelid & D. Flickinger. 2012. Who did what to whom?: a contrastive study of syntacto-semantic dependencies. In *Proceedings of the sixth linguistic annotation workshop*, 2–11. Association for Computational Linguistics.

Jackendoff, R.S. 1992. *Semantic structures*, vol. 18. MIT press.

Katz, J.J. & J.A. Fodor. 1963. The structure of a semantic theory. *Language* 39(2). 170–210.

Kilgarrif, Adam. 1998. Senseval: An exercise in evaluating word sense disambiguation programs. In *Proc. of the first international conference on language resources and evaluation*, 581–588.

Kučera, H. & W.N. Francis. 1967. *Computational analysis of present-day american english*. Dartmouth Publishing Group.

Landes, Shari, Claudia Leacock & Christiane Fellbaum. 1998. Building semantic concordances. In Fellbaum (1998) chap. 8, 199–216.

Liu, Dong C & Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical programming* 45(1). 503–528.

MacKinlay, A., R. Dridan, D. McCarthy & T. Baldwin. 2012. The effects of semantic annotations on precision parse ranking. In *Sem 2012: The first joint conference on lexical and computational semantics-*, vol. 2, 228–236. Association for Computational Linguistics.

Malouf, R. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the sixth conference on natural language learning (conll-2002)*, 49–55.

Marcus, M.P., M.A. Marcinkiewicz & B. Santorini. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational linguistics* 19(2). 313–330.

McCallum, Andrew Kachites. 2002. Mallet: A machine learning for language toolkit .

McCarthy, D., R. Koeling, J. Weeds & J. Carroll. 2004. Finding predominant word senses in untagged text. In *Proceedings of the 42nd annual meeting on association for computational linguistics*, 279. Association for Computational Linguistics.

Miller, George A., Claudia Leacock, Randee Tengi & Ross T. Bunker. 1993. A semantic concordance. In *Proceedings of the workshop on human language technology* HLT '93, 303–308. Stroudsburg, PA, USA: Association for Computational Linguistics. doi:10.3115/1075671.1075742.

Miller, Scott, Heidi Fox, Lance Ramshaw & Ralph Weischedel. 2000. A novel use of statistical parsing to extract information from text. In *Proceedings of the 1st north american chapter of the association for computational linguistics conference* NAACL 2000, 226–233. Stroudsburg, PA, USA: Association for Computational Linguistics. `http://dl.acm.org/citation.cfm?id=974305.974335`.

Niles, I. & A. Pease. 2003. Mapping WordNet to the SUMO ontology. In *Proceedings of the ieee international knowledge engineering conference*, 23–26.

Nivre, J. 2006. *Inductive dependency parsing*. Springer.

Oepen, S., D. Flickinger, K. Toutanova & C.D. Manning. 2004. Lingo Redwoods. *Research on Language & Computation* 2(4). 575–596.

Oepen, S. & J.T. Lønning. 2006. Discriminant-based MRS banking. In *Proceedings of the 5th international conference on language resources and evaluation (lrec 2006)*, .

Oepen, Stephan. 2001. [incr tsdb()] — competence and performance laboratory. User manual. Technical report Computational Linguistics, Saarland University Saarbrücken, Germany.

Pollard, C.J. & I.A. Sag. 1994. *Head-driven phrase structure grammar*. University of Chicago Press.

Pustejovsky, J. 1991. The generative lexicon. *Computational linguistics* 17(4). 409–441.

Resnik, Philip. 1998. *WordNet and class-based probabilities* 305–332. In Fellbaum (1998).

Sag, I., T. Wasow & E. Bender. 2000. Syntactic theory: A formal introduction. *Computational Linguistics* 26(2). 295–295.

Schuler, K.K. 2005. VerbNet: A broad-coverage, comprehensive verb lexicon .

Siegel, Melanie & Emily M. Bender. 2002. Efficient deep processing of japanese. In *Proceedings of the 3rd workshop on asian language resources and international standardization at the 19th international conference on computational linguistics*, Taipei, Taiwan.

Toutanova, K., C. Manning, S. Shieber, D. Flickinger & S. Oepen. 2002. Parse disambiguation for a rich HPSG grammar. *First Workshop on Treebanks and Linguistic Theories (TLT2002), 253-263* .

Toutanova, Kristina, Christopher D Manning, Dan Flickinger & Stephan Oepen. 2005. Stochastic hpsg parse disambiguation using the redwoods corpus. *Research on Language and Computation* 3(1). 83–105.

Wierzbicka, A. 1996. *Semantics: Primes and universals*. Oxford University Press, USA.

Ytrestøl, G., S. Oepen & D. Flickinger. 2009. Extracting and annotating Wikipedia subdomains. In *Proceedings of the 7th international workshop on treebanks and linguistic theories*, 185–197.

Zhang, Y. & H.U. Krieger. 2011. Large-scale corpus-driven PCFG approximation of an HPSG. In *Proceedings of the 12th international conference on parsing technologies*, 198–208. Association for Computational Linguistics.

# Appendix A

## DMRS SCHEMA

```
<!ELEMENT dmrs-list (dmrs)*>
<!-- shares several components with rmrs -->
<!-- features and values now correspond to 09/02 ERG SEMI -->
<!ELEMENT dmrs (node|link)*>
<!ATTLIST dmrs
          cfrom CDATA #REQUIRED
          cto   CDATA #REQUIRED
          surface   CDATA #IMPLIED
          ident     CDATA #IMPLIED >

<!ELEMENT node ((realpred|gpred), sense?, sortinfo)>
<!ATTLIST node
          nodeid CDATA #REQUIRED
          cfrom CDATA #REQUIRED
          cto   CDATA #REQUIRED
          surface   CDATA #IMPLIED
            base      CDATA #IMPLIED
          carg CDATA #IMPLIED >

<!ELEMENT realpred EMPTY>

<!ATTLIST realpred
          lemma CDATA #REQUIRED
          pos (v|n|j|r|p|q|c|x|u|a|s) #REQUIRED
          sense CDATA #IMPLIED >

<!ELEMENT gpred (#PCDATA)>

<!ELEMENT sortinfo EMPTY>
<!ATTLIST sortinfo
```

```
          cvarsort (x|e|i|u) #IMPLIED
          num  (sg|pl|u) #IMPLIED
          pers (1|2|3|1-or-3|u) #IMPLIED
          gend (m|f|n|m-or-f|u) #IMPLIED
          sf (prop|ques|comm|prop-or-ques|u) #IMPLIED
          tense (past|pres|fut|tensed|untensed|u) #IMPLIED
          mood (indicative|subjunctive|u) #IMPLIED
          prontype (std_pron|zero_pron|refl|u) #IMPLIED
          prog (plus|minus|u) #IMPLIED
          perf (plus|minus|u) #IMPLIED
          ind  (plus|minus|u) #IMPLIED >

<!ELEMENT sense EMPTY>
<!ATTLIST sense
          wn CDATA #REQUIRED
          lexsn CDATA #REQUIRED
          wn_lemma CDATA #IMPLIED
          offset CDATA #IMPLIED >

<!ELEMENT link (rargname, post)>
<!ATTLIST link
          from CDATA #REQUIRED
          to   CDATA #REQUIRED >

<!ELEMENT rargname (#PCDATA)>

<!ELEMENT post (#PCDATA)>
```

# Appendix B

## GOLD SENSE MAPPING

Treebanking SemCor data set was done with the intention of combining sense annotations with the ERG feature structures. However, SemCor sense tags were not matched to the ERG parses prior to this work. Here, I will describe the methodology I used to project SemCor sense annotations onto the active ERG parses. The resulting sense-annotated SemCor data set is available for download from `https://sites.google.com/site/zpozen/clms-thesis` and should be of use to other researchers interested in exploring lexical semantics and the ERG.

The gold sense annotations were not used for the parse re-ranking experiments presented in this thesis (and are therefore relegated to this appendix). The reason for this decision was a subtle but significant problem with sense-tagging inactive (dispreferred) parses. SemCor annotations are POS-tagged and, although not parsed, the sense annotations clearly result from the annotator having a specific sentence structure in mind. It is reasonable to assume that that implicit SemCor parse is very similar to the parse selected by Redwoods annotators as well. The same cannot be said of the parses rejected by Redwoods treebank annotators. Therefore, matching them with SemCor sense tags would be unsound and lead to systematic differences between models incorporating word senses for active and inactive parses, making for a problematic evaluation environment.

Figure B.1 illustrates the SemCor annotation for a single sentence. SemCor annotations are formatted in SGML, tokenized and POS-tagged. Most open-class words (such as *had* and *lunch* in this example) are manually disambiguated and tagged with their WordNet senses. *wn_lemma* attribute contains the WordNet lemma for the tagged word. *wn* attribute is the WordNet sense number for this lemma (by frequency, with "1" being the most frequent sense). *lexsn* represents the WordNet sense key, taking the

```
<s snum=53>
<wf cmd=ignore pos=PRP>He</wf>
<wf cmd=done pos=VB lemma=have wnsn=6 lexsn=2:34:00::>had</wf>
<wf cmd=done pos=VB lemma=lunch wnsn=1 lexsn=2:34:00::>lunch</wf>
<wf cmd=ignore pos=IN>with</wf>
<wf cmd=done rdf=person pos=NNP lemma=person wnsn=1 lexsn=1:03:00:: pn=person>Pauling</wf>
<punc>.</punc>
</s>
```

Figure B.1: SemCor annotation for sentence *He had lunch with Pauling*.

following form: *lemma%ss_type:lex_filenum:lex_id:head_word:head_id*. Two notable fields making up the sense key are the *ss_type*, a one-digit integer representing the part of speech, and *lex_filenum*, a two-digit decimal integer corresponding to the WordNet semantic file (SF), introduced in Chapter 2.[1] In addition to open-class words, SemCor annotation also tags named entities and categorizes them as *person*, *location*, *group* or *other* (Landes et al., 1998, p207). For example, in the sentence in figure B.1, *Pauling* is tagged as *person*. These named entity classes are also linked to WordNet senses.

The ERG MRS and DMRS/DMRX formats were introduced in Chapter 2. MRS sentence representations exported into *DMRX format* are used as input to my mapping procedure.

There are several challenges to matching the ERG DMRS sentence represenations to the SemCor. They start with the differences in tokenization and POS tagging: for example, the ERG considers punctuation marks part of the surface token they follow, while SemCor strips them. Differences between the WordNet and the ERG multiword expression coverage and analysis approach are also significant. In addition, as described in the Background chapter, some common words, such as copulas, are treated as semantically empty by the ERG, but considered semantically contentful by WordNet and sense-tagged in SemCor. The semantic decomposition in the ERG creates another

[1]The rest of *lexsn* format is described in detail as part of WordNet documentation.

problem, by positing several predicates for the same surface token: which of them should be tagged with the WordNet sense?

Figure B.2 schematically illustrates several cases in single-word predicate alignment between ERG DMRS and SemCor. Figure B.2a corresponds to the simplest case, when ERG and SemCor predicates align completely: the same surface word gets exactly one ERG predicate and exactly one SemCor sense tag.

In B.2b, the ERG posits a predicate that does not have a SemCor sense annotation. This happens most often when the ERG predicate is posited for a closed-class word (a quantifier, preposition or a conjunction), since SemCor only tags open-class words.

The case in B.2c occurs when SemCor sense-tags a word that is treated as semantically empty by the ERG analysis, such as a copula *be*, most frequently annotated as WordNet sensekey *be%2:42:03::*. This situation also occurs when SemCor tokenization separates the negation clitic *n't* into a separate word and tags it, but the ERG creates a special gpred, *neg_rel*, with a surface span including the entire orthographic token (such as *wouldn't*).

Finally, B.2d represents ERG semantic decomposition: certain lexical items are split into multiple ERG predicates, while only getting one WordNet sense. For example, *today* always generates an additional gpred *time_n_rel*, in addition to a real predicate with the lemma *today*. SemCor sense annotation is completely flat and does not tag a single surface item with more than one sense.

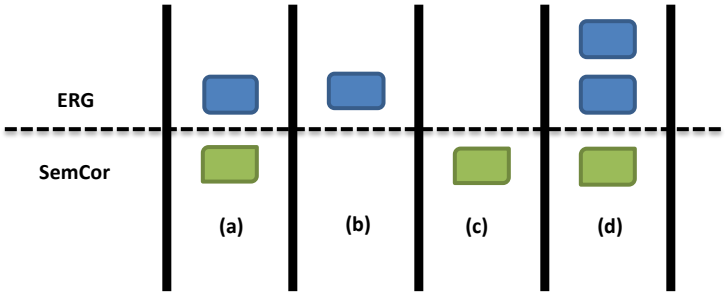


Figure B.2: Single-word predicates in SemCor and the ERG predicates

Figure B.3 represents two cases where multiword expression analyses or coverage

differ between the ERG and the SemCor WordNet-based annotation. In B.3a, WordNet applies a single sense annotation to a multiword expression, whereas the ERG treats it compositionally, as separate predicates. Some cases when SemCor decides to identify an idiom are questionable (e.g., *red clay*, or *mental picture*). In other cases, like *over and over* or *in the end* it seems like the ERG coverage may have to be extended to include these expressions.

Figure B.3b shows the reverse situation: the ERG recognizes a multiword expression as a single predicate, but WordNet treats it compositionally: for instance *a little* is a single ERG predicate, while the SemCor annotator only tagged *little* separately, despite the idiom being known to WordNet.

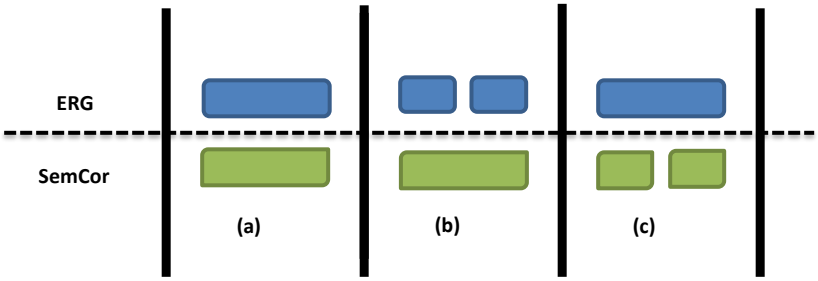


Figure B.3: Multiword expressions in SemCor annotations and the ERG

To summarize, SemCor sense annotation is flat and gappy, while the ERG has predicate overlaps, and in principle only has gaps for the semantically empty elements. Keeping these properties in mind is essential for understanding the mapping procedure.

The actual mapping algorithm proceeds as follows:

1. Add all SemCor sense-tagged words into an ordered collection $V_s$.

2. Add all ERG predicates with unique surface spans into a collection $V_e$. If two predicates cover the same surface span, the following deterministic procedure is employed to select one of them:

   (a) If one predicate is a *realpred* and another a *gpred*, a realpred is selected.

(b) If both predicates are *realpred* (this happens with compound word decomposition, e.g. for words like *pokerface* or *misunderstand*), the predicate with a higher id is selected, because compound English constructions tend to be head-final (*face* and *understand* in our examples).

(c) If both predicates are gpreds:

i. If a non-quantifier *gpred* and any quantifier relation *gpred* (ending in "_q_rel") are present, a non-quantifier is selected.

ii. Else, the *gpred* that has a lower overall observed occurrence count in the Redwoods SemCor corpus is selected. This is motivated by the idea that a less frequent *gpred* carries more information, making it a better candidate for associating with sense annotation.

iii. If the occurrence counts are the same, the *gpred* whose name precedes the other alphabetically is selected. This is an unmotivated, but deterministic tie-breaking fallback.

3. All elements in $V_e$ are marked as untagged.

4. For each $V_s$ item, $V_{si}$, the procedure scans $V_e$ to find the first untagged element, $V_{ej}$, such that **predicate_match**($V_{si}$, $V_{ej}$) = *True*. and marks $V_{ej}$ as tagged. At this point all the sense information for ($V_{si}$ is attached to $V_{ej}$).

The **predicate_match()** procedure determines whether an ERG and a SemCor predicate are considered a match. It takes the following as input:

- $v_s$: the SemCor predicate surface string given by the value of *wf* SemCor element.

- $v_e$: the ERG predicate surface string, extracted from the surface sentence starting at "cto" and ending at "cfrom" DMRX attributes.

- *sent_erg-mod*: ERG surface string from $v_e$ until the end of the sentence, converted to lowercase, with punctuation marks removed.

- *pred_erg*: the type of the ERG predicate (*realpred* or *gpred*).
- *pos_erg*: if *pred_erg* is *realpred*, its part of speech tag (undefined otherwise).

**predicate_match()** is defined as follows:

1. $v_s$ and $v_e$ are converted to lowercase and all punctuation marks removed, to create $v_{s\text{-}mod}$ and $v_{e\text{-}mod}$.
2. If $v_{s\text{-}mod}$ and $v_{e\text{-}mod}$ are equal at this point, return *true*.
3. Else, consider $v_s$ a phrasal verb whose head verb matches $v_e$ if all of the following is true:
   - $v_s$ contains the underscore character (the connector of multiword expressions in WordNet)
   - The $v_s$ substring following the underscore matches one of the 91 common prepositions
   - *pred_erg* is *realpred*
   - *pos_erg* is *verb*
   - $v_{e\text{-}mod}$ is the prefix of *sent_erg-mod*
4. If all the above phrasal verb conditions are true, return *true*.
5. Else, return *false*.

I'd like to highlight a few points about the **predicate_match()** procedure described above. It encodes a set of empirically discovered heuristics and is obviously very language-specific. The way it handles tokenization differences between the ERG and SemCor is by stripping all punctuation when comparing surface forms. It also does not generally consider part of speech tags when matching elements, making it robust against POS tag mismatches. The only part of the matching process that concerns itself with the POS tags is phrasal verb identification.

The decision to only handle phrasal verbs, rather than extend the same methodology to other multiword expressions (MWEs) distinguished by SemCor but compositional in the ERG analysis, merits an explanation. The motivating idea was that while sense annotation is appropriate to attach to the head verb of a phrasal verb MWE, it is unclear which element of an MWE is the head and should be tagged in the general case. Luckily, phrasal verbs constituted a very large portion of the MWEs in the SemCor corpus, so the special handling for them has paid off. The bulk of the SemCor wordforms that didn't get matched to ERG predicates as a result of this procedure were MWEs missing from the ERG lexicon as a result of lower coverage or compositional analysis.

To store the mapped entries, I extended the DTD schema for DMRX format (described in Appendix A) by adding a *sense* element, to be optionally included by *realpred* or *gpred* nodes. The *sense* element has five optional CDATA-typed attributes. For of those, *wn_lemma*, *lexsn*, *wn*, correspond to SemCor SGML wordform element attributes with the same names and are applied to DMRS nodes as part of the gold sense matching process. The *wn_lemma* attribute containing the WordNet lemma is usually, though not always, the same as the ERG lemma. The fourth attribute, *offset*, representing WordNet synset offset for the disambiguated word, is reserved for future work and not generated by the SemCor sense mapping process described here. Finally, the *sf* attribute, represents the semantic file. It is also not generated at this point, and is intended for first sense mapping, described in Chapter 4. Figure B.4 shows the relevant excerpt from the updated schema. The complete DTD for the updated schema is listed in the Appendix A.

Figure B demonstrates the tagged annotation produced as a result of the process described in this appendix (with "node" and "sortinfo" elements removed for brevity).

The resulting mapping of the XML DMRS format is published at the thesis website. The results of the mapping are summarized in table B.1.

```
<!ELEMENT dmrs-list (dmrs)*>
<!ELEMENT dmrs (node|link)*>
<!ELEMENT node ((realpred|gpred), sense?, sortinfo)>
<!ELEMENT sense EMPTY>
<!ATTLIST sense
          wn_lemma  CDATA #IMPLIED
          lexsn CDATA #IMPLIED
          wn    CDATA #IMPLIED
          offset    CDATA #IMPLIED
          sf CDATA #IMPLIED >
```

Figure B.4: Optional *sense* element DTD schema for DMRX.

```
<gpred>pron_rel</gpred>
<gpred>pronoun_q_rel</gpred>
<realpred lemma='have' pos='v' sense='1'/><sense wn='6' lexsn='2:34:00::' wn_lemma='have'/>
<gpred>udef_q_rel</gpred>
<realpred lemma='lunch' pos='n' sense='1'/><sense wn='1' lexsn='2:34:00::' wn_lemma='lunch'/>
<realpred lemma='with' pos='p'/>
<gpred>proper_q_rel</gpred>
<gpred>named_rel</gpred><sense wn='1' lexsn='1:03:00::' wn_lemma='person'/>
```

Figure B.5: Gold sense tags for the example sentence

| | |
|---|---|
| Total unique span ERG predicates | 38,381 |
| Total tagged SemCor tokens | 19,669 |
| Total matched SemCor tokens | 18,451 |
| Percent matched SemCor tokens | 93.80% |

Table B.1: Summary of SemCor annotation mapping to Redwoods