It's Only Morpho-Logical: Modeling Agreement in Cross-Linguistic Dependency Parsing

Matthew Hohensee

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

Master of Science

University of Washington

2012

Emily M. Bender, Chair

Fei Xia

Program Authorized to Offer Degree:
Computational Linguistics

University of Washington

Abstract

It's Only Morpho-Logical: Modeling Agreement in Cross-Linguistic Dependency Parsing

Matthew Hohensee

Chair of the Supervisory Committee:
Associate Professor Emily M. Bender
Department of Linguistics

I propose a linguistically motivated set of features to model morphological agreement and add them to MSTParser, a graph-based dependency parser (McDonald et al., 2006). Compared to the parser's built-in morphological features, the new feature set is much smaller and more accurate. Results across 21 treebanks containing varying amounts of morphological annotation demonstrate increases in accuracy of up to 5.3% absolute. Experiments are performed to investigate exactly how the features enhance performance. While some of the improvement results from the feature set capturing information unrelated to morphology, there is still significant improvement, up to 4.6% absolute, due to the agreement model.

This thesis includes background on morphological agreement and dependency parsing, details on MSTParser and modifications made to it, information about the treebanks collected and the steps taken to normalize them, and descriptions of experiments and results.

TABLE OF CONTENTS

Page

List of Figures iii

List of Source Code Listings iv

List of Tables v

Chapter 1: Introduction 1

Chapter 2: Background 3
2.1 Morphology and Agreement 3
2.2 Dependency Grammar 5

Chapter 3: Literature Review 8
3.1 Dependency Parsing 8
3.2 MSTParser 9
3.3 Incorporating Morphology and Agreement into NLP systems 12
3.3.1 CoNLL Shared Tasks 12
3.3.2 Further Research on Morphology and Dependency Parsing . . 15
3.3.3 Agreement Modeling in Other Tasks 17
3.3.4 Summary 18

Chapter 4: Methodology 20
4.1 Implementation of Agreement Model 20
4.1.1 Motivation and Theory 20
4.1.2 Agreement Model Features 21
4.2 Data Collection and Preparation 22
4.3 Experimental Setup 27

Chapter 5: Experiments and Results 29
5.1 Overall Results 29
5.2 Improvement vs. Dataset Size 34
5.3 Gold vs. Automatically Predicted Annotations 36
5.4 PPL feature 36
5.5 Weights of Original and Agreement Features 41
5.6 Including All vs. Some Attributes 42
5.7 Summary 45

Chapter 6: Future Work and Conclusion 47
6.1 Future Work 47
6.2 Conclusion 47

References 49

Appendix A: Implementation of Agreement Features 58

Appendix B: Original to Universal POS Tag Mappings 61

Appendix C: English POS Tag to Morphological Annotation Mapping 65

Appendix D: Morphological Information in Treebanks 66

Appendix E: Underscore/Root Feature Modification to MSTParser 71

Appendix F: Raw Data 73

Appendix G: PPL Modification to MSTParser 75

LIST OF FIGURES

Figure Page

5.1 Accuracy of Feature Configurations by Treebank 32
5.2 Error Reduction by Treebank 33
5.3 Accuracy vs. Dataset Size, Czech 34
5.4 Accuracy vs. Dataset Size, English 35
5.5 Accuracy of Feature Configurations by Treebank, with PPL Feature . . . 39
5.6 Error Reduction by Treebank, with PPL Feature 40
5.7 Unlabeled Accuracy on Czech with Various Attributes Included 44

LIST OF SOURCE CODE LISTINGS

Listing Page

A.1 Original Morphological Features 58
A.2 Unlabeled Agreement Features 59
A.3 Labeled Agreement Features 60

E.1 Underscore Modification 71
E.2 Root Feature Modification 72

G.1 PPL Feature 75

LIST OF TABLES

Table Page

3.1 Original MSTParser Feature Templates 10

4.1 Agreement Model Feature Templates 22
4.2 Treebanks Used 23
4.3 CoNLL Treebank Format 24
4.4 Sample Sentence and Agreement Features Generated 27

5.1 Results on All Treebanks 30
5.2 Effect of Automatic Annotations on Parsing Accuracy 37
5.3 Highly-Weighted Features with and without PPL Feature 38
5.4 Results on All Treebanks, with PPL Feature 41
5.5 Correlation of Error Reduction with Morphological Information 42
5.6 Highly-Weighted Features in Each Feature Configuration 43
5.7 Number of Highly-Weighted Agreement Features by Attribute 45

B.1 Universal POS Tag Mapping for Finnish Treebank 61
B.2 Universal POS Tag Mapping for Ancient Greek and Latin Treebanks . . 62
B.3 Universal POS Tag Mapping for Hebrew Treebank 62
B.4 Universal POS Tag Mapping for Hindi Treebank 63
B.5 Universal POS Tag Mapping for Italian Treebank 63
B.6 Universal POS Tag Mapping for Tamil Treebank 64

C.1 POS Tag to Morphological Annotation Mapping for English Treebank . . 65

D.1 Summary of Morphological Information in Each Treebank 66

E.1 Effect of Underscore/Root Feature Modification 72

F.1 Accuracies on All Treebanks 73
F.2 Accuracies on All Treebanks, with PPL Feature 74

ACKNOWLEDGMENTS

I would like to thank Emily Bender for her invaluable guidance and encouragement; Fei Xia for her helpful comments; David Brodbeck for technical support; all the kind souls who assisted me in gathering data, particularly Maite Oronoz and her colleagues at the University of the Basque Country; and finally, Katie, for her endless support and patience.

Chapter 1
INTRODUCTION

NLP applications can be categorized based on the degree to which they incorporate linguistic knowledge, which is often assumed to correlate with their degree of language-independence. They may be language-specific, leveraging deep knowledge of the language for which they are designed, often via hand-crafted grammars and models. Or, they may be language-independent, primarily statistical in nature and incorporating a minimum of linguistic intuition. However, building a system without the use of any specific linguistic details does not necessarily guarantee its language-independence. Even linguistically naïve systems can involve design decisions which in fact bias the system towards languages with certain properties (Bender, 2011).

Conversely, it is often taken for granted that using linguistic information necessarily makes a system language-specific. But it is possible to design a linguistically intelligent system without tuning it to a specific language. This could be done by modeling cross-linguistic phenomena at a high level. Such a system would still be language-independent; it would not require any knowledge or modeling of specific languages, but it would ideally use linguistic knowledge to make the most of the available data.

Morphological agreement, the appearance of specific morphological attributes on words which are syntactically related in certain ways, is a common phenomenon in many languages. Because the information encoded in agreement – syntactic relationships between words – could be useful in parsing, I decided to try to leverage this information in the context of dependency parsing. To do this, I developed a set

of machine learning features to model agreement. These features allow the parser to make parsing decisions based on the presence or absence of agreement, rather than on the morphology of individual words or pairs of words. Furthermore, since agreement appears cross-linguistically, the features are applicable to a diverse set of languages.

Most data-driven dependency parsers are meant to be language-independent. They do not use any information that is specific to the language being parsed, and they often rely heavily on n-grams, or sequences of words and part-of-speech tags, to make parsing decisions. In this thesis, I describe the modification of such a parser, MSTParser (McDonald et al., 2006), to incorporate a model of morphological agreement. This modification improves parsing performance across a variety of languages by making better use of agreement relationships reflected in morphological annotations. I test the improved system on a variety of treebanks and find that, compared to the original approach, it is faster, more compact, and more accurate.

The new feature set improves accuracy by up to 5.3% absolute, depending on the treebank. Several experiments indicate that while part of this improvement seems to be due to factors unrelated to morphology, much of it can be ascribed to the agreement model. Since the modified parser uses a much smaller feature set than does the original, feature counts are also lower, and run times faster.

In Chapter 2, I present background information on morphological agreement and dependency grammar. Chapter 3 surveys the literature on dependency parsing and the use of morphological information therein and presents a description of MSTParser. Chapter 4 describes in detail the modifications made to MSTParser, the data I collected and the steps taken to prepare it, and the experimental setup. In Chapter 5, I describe the experiments run and present the results. Chapter 6 presents some potential directions for future work and concludes.

Chapter 2
BACKGROUND

This chapter consists of short overviews of two topics at the heart of this work: morphological agreement and dependency grammar.

2.1 Morphology and Agreement

Many of the world's languages show some level of morphological agreement. That is, syntactic relationships between words are reflected in their morphology – prefixes, suffixes, or other inflection. For instance, a noun and its determiner may both reflect morphologically whether the noun is singular or plural in number; if this happens, they are said to agree for number, or to be marked for number. Relationships such as this can involve nouns, verbs, adjectives, determiners, and occasionally adverbs, and can represent linguistic attributes such as person, number, gender, case, definiteness, and others (Corbett, 2006).

A note on terminology: henceforth, the terms *attribute* and *value* will be used to refer to aspects of this phenomenon. Linguistic properties will be described in terms of *attributes*; a given word has one *value* for each attribute. For example, the English word *dogs* has the value "plural" for the attribute "number". The term *feature* will be reserved for use in the machine learning sense.

The frequency of agreement relationships in a language varies across a wide spectrum. At one end of this spectrum are analytic languages, also referred to as "morphologically impoverished" or "morphologically poor". An extreme example is Chinese, which shows hardly any inflection, and no agreement whatsoever. Although it does have particles which can indicate attributes like number and aspect, these attributes

are not marked on multiple words, so no agreement can be said to take place.

English is a less extreme case. Although nouns and determiners are marked for number, and verbs (except the copula "to be") have a total of four to five possible forms, adjectives are not marked. Gender and case are marked only on pronouns, and thus cannot dictate agreement relationships.

At the other end of the spectrum are synthetic or "morphologically rich" languages. Czech, for example, has a system of seven cases and two numbers. Thus, a noun could theoretically appear in fourteen different forms depending on its number and its syntactic function in a sentence. More importantly, any determiners or adjectives referring to that noun must reflect the number and case of the noun as well.

A sample sentence in Czech, English, and Chinese demonstrates this contrast. In the Czech sentence (1), the adjective and noun agree for gender, number, and case, and the noun and verb agree for person and number. These agreement relationships are realized by suffixes. In the English translation (2), only the noun and verb agree: the noun is inherently plural third person, and the verb shows agreement by its lack of any suffix. In the Chinese version (3), there is no agreement.

(1) Zahraniční investice rostou.
foreign.FEM.PL.NOM investment.FEM.3.PL.NOM grow.3.PL.PRES

(2) Foreign investments grow.
foreign investment.3.PL grow.PL

(3) 外商 投资 增长 了。
foreign investment grow PTCL

A widely accepted generalization holds that morphologically richer languages generally exhibit more variation in word order than more analytic languages, based on the assumption that the former use inflection to mark the roles of constituents, while the latter encode this information in word order. For instance, word order in Czech is relatively free, while in English it is largely fixed. Siewierska (1998) investigated this tendency in a sample of 171 diverse languages and found that word order is generally

fixed in analytic languages, though not necessarily flexible in synthetic languages.

It is important to note that agreement can take a variety of forms. While it is often manifested as a suffix or prefix, the lack of any affix can also represent agreement, as can other inflections such as stem vowel changes. Or, the property in question may be inherent to the word; for example, non-pronominal nouns are inherently third person. That is, a word need not necessarily display any specific morphological marker in order to be said to participate in an agreement relationship. Finally, a word can be ambiguous as to its value for an attribute; e.g., in English, the second-person pronoun *you* could be in either the nominative or the accusative case, whereas the first-person pronoun *me* is necessarily in the accusative case. This is known as syncretism.

2.2 Dependency Grammar

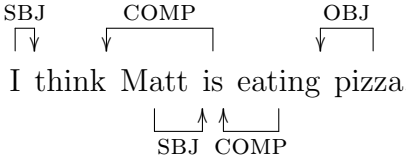
Tesnière's *Elements of Structural Syntax* (1959) is considered the starting point of modern dependency grammar. Tesnière suggested that asymmetric binary links between words could be sufficient to describe the structure of a sentence, without the need to posit any phrasal constituents. Thus, in dependency grammars, as opposed to constituency or phrase structure grammars, the only nodes are the words of the sentence. Each word is associated with another word, called its *head*, and that relation is called a *dependency*. The head may also be referred to as the *parent*, and the dependent node as the *child*; children and their children may be referred to as *descendants*, and multiple children of the same parent as *siblings*. The dependencies are often labeled with types which categorize the relationship the dependent has to its head. One word (usually the finite verb) acts as the root of the tree and has no head, or has as its head a special root node not associated with any word. An example of a dependency parse follows:

(4) Matt is eating pizza.

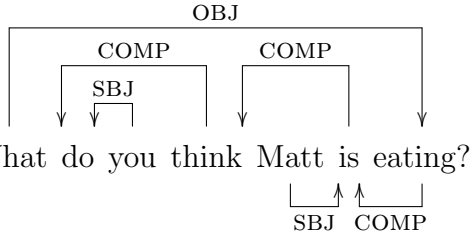
The dependency arcs in (4) indicate that the noun *Matt* is headed by the finite verb *is* in a subject relation; that the gerund *eating* is headed by *is* in a complement relation; and that the noun *pizza* is headed by *eating* in an object relation.

Dependency parsing is of particular interest to computational linguists because it provides the structure necessary for higher-level applications but incorporates a minimum of syntactic formalism. This simplifies both parsing and subsequent utilization of the parses. During parsing, the algorithm must only connect the words of the sentence, without having to construct intermediate nodes. Once the parses have been generated, many relationships can be read from them quickly, with no need for complicated processing of tree structure (Covington, 2001).

The concept of projectivity has important implications for dependency parsing. A projective parse is one in which, in the linear order of words in the sentence, each head forms a contiguous sequence with its descendants. Stated equivalently, in a projective parse, for any arc from head h to dependent d, all words between h and d in the sentence are descendants of h. (5) and (6) display parses for two similar sentences which demonstrate the contrast between projective and non-projective constructions.



(5)



(6)

(5) illustrates a projective parse; in (6), because of the wh-fronting, *eating* is separated from its dependent (*what*) by other words which are not its descendants. Thus, the parse is non-projective.

The distinction between projective and non-projective can have consequences for NLP system design, as algorithms which generate only projective parses often have lower computational complexity than those which are able to generate non-projective parses as well. For this reason, treebanks vary in whether they include non-projective parses or represent them projectively (using traces, for example). The frequency of non-projective sentences differs among languages based on the presence of syntactic movement phenomena, such as wh-fronting and focus movement in the case of English.

Chapter 3
LITERATURE REVIEW

This chapter includes surveys of the existing literature on dependency parsing and the use of morphology therein, and a description of MSTParser.

3.1 Dependency Parsing

Dependency parsers have developed along several lines. The first dependency parsers were based on grammar-driven CFG parsers, such as the CKY (Kasami, 1965; Younger, 1967) and Earley (1970) parsers; probabilistic versions of these were implemented once sufficient amounts of data became available. Eliminative or constraint-based parsers have been developed by Karlsson (1990) and Maruyama (1990). Data-driven grammar-based parsers, which induce generative probabilistic models from treebanks, were developed by Eisner (1996), Collins et al. (1999), and Samuelsson (2000).

The most recent parsers are data-driven as well, but use discriminative models induced from treebanks. These systems generally fall into two groups: transition-based and graph-based. Transition-based parsers, exemplified by MaltParser (Nivre, Hall, & Nilsson, 2006), process sentences linearly, using shift-reduce parsing and selecting the correct parse action at each step via machine learning. Graph-based parsers, such as MSTParser (McDonald et al., 2006), treat the parsing problem as a search through the forest of all possible dependency trees for the best candidate, scoring each tree based on weights learned during training.

3.2 MSTParser

MSTParser (McDonald et al., 2006) is a data-driven, graph-based system which creates a model from training data by learning weights for a set of arc-level features. It is written in Java and freely distributed as open-source software, and was the top performer in the CoNLL-X shared task on multilingual dependency parsing (Buchholz & Marsi, 2006). I chose it, rather than the transition-based MaltParser, to focus on for the present research because it generates both projective and non-projective parses natively, and because it has fewer parameters which could potentially complicate testing across a variety of languages.

Graph-based parsing is based on the idea that the union of all possible dependency parses of a sentence is equivalent to a strongly connected directed graph, where the vertices are the words of the sentence. Each parse, then, is equivalent to a spanning tree of the graph – that is, a tree which includes every vertex – rooted at the root node of the sentence. The collection of every possible such tree is referred to as the parse forest.

MSTParser uses a feature representation to find the best possible tree, or parse, for a sentence; each tree is defined via a set of arc-level features. The feature set includes various combinations of the word and POS tag of the parent and child of each dependency arc; POS tags of words between the parent and child; POS tags of the parent and child along with those of the words immediately following or preceding them; and finally, morphological attributes derived from the parent and child tokens' morphological information. An optional set of "second-order" features includes analogous information about siblings. A similar feature set is conjoined with arc labels in order to perform labeling.

Morphological features for an arc are generated by iterating over each pair in the cross product of the parent and child tokens' lists of attributes. For every such pair, thirteen groups of four features each are generated. The thirteen groups represent

Table 3.1: Original MSTParser feature templates. `hdForm` and `dpForm` are the head and dependent word forms; `hdLemma` and `dpLemma` are the lemmas (if available). `hdAtt` and `dpAtt` are the current morphological attributes; `hdIdx` and `dpIdx` are their list indices. `dir+dist` is a string encoding the direction and length of the dependency arc. Each line represents one feature.

<hdIdx>*<dpIdx>=<{hdForm|hdLemma}>(<dir+dist>)
<hdIdx>*<dpIdx>=<{dpForm|dpLemma}>(<dir+dist>)
<hdIdx>*<dpIdx>=<hdAtt>(<dir+dist>)
<hdIdx>*<dpIdx>=<dpAtt>(<dir+dist>)
<hdIdx>*<dpIdx>=<{hdForm|hdLemma}><{dpForm|dpLemma}>(<dir+dist>)
<hdIdx>*<dpIdx>=<{hdForm|hdLemma}><hdAtt>(<dir+dist>)
<hdIdx>*<dpIdx>=<{hdForm|hdLemma}><dpAtt>(<dir+dist>)
<hdIdx>*<dpIdx>=<{dpForm|dpLemma}><dpAtt>(<dir+dist>)
<hdIdx>*<dpIdx>=<{dpForm|dpLemma}><hdAtt>(<dir+dist>)
<hdIdx>*<dpIdx>=<hdAtt><dpAtt>(<dir+dist>)
<hdIdx>*<dpIdx>=<{hdForm|hdLemma}><hdAtt><dpAtt>(<dir+dist>)
<hdIdx>*<dpIdx>=<{dpForm|dpLemma}><hdAtt><dpAtt>(<dir+dist>)
<hdIdx>*<dpIdx>=<{hdForm|hdLemma}><{dpForm|dpLemma}><hdAtt><dpAtt>(<dir+dist>)

combinations of the head and child word forms/lemmas and attributes. Each of those groups, in turn, contains two subgroups, one which uses the word form and one which uses the lemma; those subgroups in turn each contain two features, one which is conjoined with the direction and distance of the dependency and one which is not. Every feature is prefixed by the indices of the head and child attributes within their respective attribute lists.[1] These features are summarized in Table 3.1.

Feature weights are determined using the Margin Infused Relaxed Algorithm, or MIRA (McDonald, Crammer, & Pereira, 2005). MIRA is an iterative learning algorithm which considers training instances consecutively, making minimal changes to the weight vector while maintaining a minimum score margin.

Decoding can be performed in several ways: with or without "second-order" (sib-

[1]While the inclusion of the list indices could potentially lead to sparseness issues, this is not a concern in practice. Attribute lists generally appear in the same order for all tokens, so each attribute only co-occurs with one index. This ordering in the data is not enforced in any way, however.

ling) features, and in projective (generating only projective parses) or non-projective (generating all parses) mode. Second-order parsing was found to perform better across multiple languages, and was used in the CoNLL shared task.

For projective parsing, Eisner's algorithm (1996) is used. This algorithm performs an exact search over all projective trees in $O(n^3)$ time for either first- or second-order parsing. For first-order non-projective parsing, the algorithm used is Chu-Liu-Edmonds (Chu & Liu, 1965; Edmonds, 1968), which performs an exact first-order search in $O(n^2)$ time, using Tarjan's implementation (1977). For second-order non-projective parsing, that algorithm becomes NP-hard. In this case, a $O(n^3)$ approximation based on the Eisner algorithm is used.

While allowing non-projective parses gives the parser the flexibility to represent any possible tree structure, doing so provides little benefit for languages which generally only require projective trees (McDonald, Pereira, et al., 2005). In fact, the Chu-Liu-Edmonds algorithm actually performs slightly worse than the Eisner algorithm on projective data, presumably because it does not necessarily prefer projective to non-projective parses. In the CoNLL-X shared task, the projective parsing algorithm was used for six languages (Arabic, Bulgarian, Chinese, Spanish, Swedish, and Turkish), based on performance on development data (McDonald et al., 2006).

The best parse of the sentence is the maximum spanning tree (MST) of the graph: the spanning tree with the highest score according to the learned feature weights. Once the best parse has been found, the dependency arcs are labeled using the weights in conjunction with a first-order Markov model and Viterbi's algorithm (McDonald et al., 2006).

3.3 Incorporating Morphology and Agreement into NLP systems

3.3.1 CoNLL Shared Tasks

The CoNLL-X (Buchholz & Marsi, 2006) and CoNLL 2007 (Nivre et al., 2007) shared tasks focused on multilingual dependency parsing. In each, participants developed parsers which predicted dependency arcs and labels for POS-tagged and morphologically analyzed data, and tested them on datasets representing a variety of languages provided by the organizers.

The CoNLL data included, for each token, the word form, POS tag, head, and arc label, and optionally, a lemma, a second POS tag, and a list of morphological attributes. The datasets varied in whether they included all, some, or none of this optional information. The morphological attributes, if present, were either atomic (e.g., `NOM`) or attribute-value pairs (e.g., `case=NOM`), depending on the treebank. Systems were scored separately for unlabeled (arcs only) and labeled (arcs and relation types) accuracy. The 2006 task included thirteen languages: Arabic, Bulgarian, Czech, Chinese, Danish, Dutch, German, Japanese, Portuguese, Slovene, Spanish, Swedish, and Turkish. The 2007 task included ten languages representing a more diverse set of language families: Arabic, Basque, Catalan, Chinese, Czech, English, Greek, Hungarian, Italian, and Turkish.

The coordinators of the 2006 shared task noted that performance on each language was largely predicted by the details of the training data: average performance was higher on larger data sets, or those drawn from more restricted domains. The participant ranking did not vary considerably across languages; certain systems performed globally better than others (Buchholz & Marsi, 2006). The two top performers in the task were MSTParser and the transition-based MaltParser. Despite their radically different approaches to the parsing problem, average scores over all languages for the two parsers were very similar.

The results of the 2007 shared task were somewhat different, perhaps due to

efforts to normalize training data and include a wider variety of languages (Nivre et al., 2007). Grouping the datasets by average score over all participants produced three clusters which aligned roughly, and inversely, with morphological complexity. Relatively analytic languages (Catalan, Chinese, English, and Italian) had the highest average scores. Among more inflected languages, agglutinative languages (Hungarian and Turkish) had moderate scores; and fusional languages (Arabic and modern Greek) had the lowest scores.[2] The organizers wrote that "the most difficult languages are those that combine a relatively free word order with a high degree of inflection" (p. 927).

The participating systems handled morphological information in the treebanks in a variety of ways. Several of the systems ignored it altogether (Johansson & Nugues, 2006; Wu et al., 2007; Dreyer et al., 2006; Liu et al., 2006). Several others used the entire list of morphological attributes as an atomic attribute, similar to a fine-grained POS tag (Chang et al., 2006; Titov & Henderson, 2007; Chen et al., 2007). The most common approach was to split the list into attributes and generate machine-learning feature based on each (Nivre, Hall, Nilsson, Eryiğit, & Marinov, 2006; Carreras et al., 2006; Yuret, 2006; Duan et al., 2007; Nguyen et al., 2007). Another common approach was to generate machine-learning features based on each pair of attributes in the cross product of the lists of a potential head and dependent (McDonald et al., 2006; Riedel & Clarke, 2006; Nakagawa, 2007; Chen et al., 2007). Note that these approaches are all language-independent; they are applicable to any treebank which includes lists of morphological attributes for each token, and they do not require any knowledge of what the attributes represent.

Other approaches used morphological information in more linguistically intelligent, and language-specific, ways. A few systems used it to disambiguate function words

[2]Though it is a fusional language, Czech fell into the "moderate" category, presumably due to its significantly greater amount of training data. Basque, an agglutinative language, fell into the "low" category, probably because it had the least training data.

(Bick, 2006; Corston-Oliver & Aue, 2006), or to pick out finite verbs (Carreras et al., 2006). Schiehlen and Spranger (2007) developed a parser whose feature representation did not incorporate morphological attributes as such; instead, they distributed selected attributes to other fields in order to capture specific information. This was done using rules specific to each language: for instance, case information was used to to create a fine-grained POS tag in languages where case was deemed relevant to determining the dependency relation, and in Turkish, morphological information indicating the semantic class of a word was used as a lemma.

Only one system (Attardi et al., 2007) appears to have modeled agreement explicitly, by generating a "morphological agreement" feature whenever two tokens possessed the same value for the same linguistic attribute. The authors note accuracy improvements of about 0.4 to 0.5% for Italian, and about 0.8% for Catalan, using a transition-based parser.[3] Only gender and number agreement are mentioned, and it is unclear whether they modeled person or case agreement in other languages which were part of the shared task (e.g., Czech). It appears that their system took into account only relationships in which tokens agree for an attribute, and not those in which they disagree, and it is unclear whether they modeled agreement language-independently, or by creating a specific model for each language.

Many of the participants mentioned plans for future work involving making better use of the morphological information (when present) in treebanks. The developers of MaltParser wrote that "the development of parsing methods that are better suited for morphologically rich languages with flexible word order appears as one of the most important goals for future research in this area" (Hall et al., 2007).

[3]All accuracies are absolute unless stated otherwise.

3.3.2 Further Research on Morphology and Dependency Parsing

Research outside the CoNLL shared tasks has produced similar results, and the potential of morphological information to improve parsing performance has been noted repeatedly.

This potential has been documented in numerous experiments on morphologically rich languages using MaltParser and including various types of morphological attributes as machine learning features (without generating any higher-level features such as agreement). Nivre et al. (2008) found that in Russian, a highly inflected language similar to Czech, "morphological features are crucial for obtaining good parsing accuracy"; they used a large set of morphological attributes, including person, gender, number, case, animacy, tense, mood, and voice, as features. In similar work on Swedish, Øvrelid and Nivre (2007) came to the same conclusion, using attributes such as definiteness, animacy, case, and semantic class to achieve accuracy improvements of around 1%.

Bengoetxea and Gojenola (2010), working with Basque, compared different feature representations of the same morphological information: treating the entire attribute list as an atomic feature vs. treating each attribute separately. They achieved the best results by using three features for each token: case, subordination (indicating the presence of a relative clause), and all other attributes grouped together as an atomic feature. This resulted in a labeled accuracy improvement of 0.8%.

Nivre (2009) also found that, when parsing Bangla, Telugu, and Hindi with MaltParser and various feature configurations, including all attributes as features increased accuracy substantially. Ambati et al. (2010) experimented with different feature sets for parsing Hindi with MaltParser. They noted a substantial increase in accuracy when adding case features for nouns and tense/aspect/modality features for verbs. However, including person, number, and gender features did not help. This was attributed to the complicated agreement system in that language.

Highly agglutinative languages present special challenges for dependency parsing. In Turkish, for example, due to the many possible layers of inflectional morphology, a single highly inflected word can carry the information of several phrases, or an entire sentence, of an analytic language. Because of this complexity, parsing is nearly impossible without the use of morphological analysis to break down words into inflectional groups. These groups can then be used either as the basic tokens of parsing, or to generate features capturing morphological attributes. Eryiğit et al. (2008) found that once these groups have been identified, the inclusion of features encoding case and possessive agreement markers greatly improved accuracy when using a classifier-based parser, and that person and number agreement markers were important as well. (Again, agreement between tokens was not modeled explicitly; the features used represented only the morphology of single tokens.)

Goldberg and Elhadad (2009), investigating the performance of both MSTParser and MaltParser on Hebrew, ran parsing experiments on data with both gold standard (oracle) and automatically predicted token segmentation, POS tagging, and morphological information. Each of these configurations was tried with and without morphological features including gender, number, and definiteness. Adding these features improved the performance of MSTParser slightly (less than 1%) when training on the gold standard data, but decreased it (around 2%) when the automatically tokenized and tagged data was used. MaltParser performance improved slightly (less than 1%) in both cases.

In subsequent work, the same authors achieved substantial improvements in the performance of their own transition-based parser by adding agreement features. In Hebrew, adjectives and nouns must agree for gender and number. Therefore, a gender-agreement feature was added when a noun-adjective pair agreed for gender; similarly for a number-agreement feature in the case of number agreement. Adding these features improved accuracy by 0.5-1.0% on both gold standard and automatically tagged data (Goldberg & Elhadad, 2010). These results, taken in conjunction with

those in the previous paper, suggest that modeling agreement, even for only one type of relation (noun-adjective in this case), may be more robust and less sensitive to tagging errors than are other approaches to morphological information.

Marton et al. (2010, 2011), experimenting with different feature sets for parsing Arabic with MaltParser, also tested the effects of including gold-standard vs. automatically predicted morphological features. They found that while performance was improved in both cases, the most useful attributes differed: person, number, gender, and definiteness were most helpful when using automatic tags, whereas case and state were most helpful when using gold tags.

Working in rule-based constituency parsing, Tsarfaty and Sima'an (2010) labeled constituents in a Hebrew treebank with gender, definiteness, and case attributes. They experimented on this data using a number of parsing models and configurations, and achieved state-of-the-art results when including the morphological information. Cowan and Collins (2005) took a similar approach, using number, person, gender, and verb mode and tense to generate finely-grained POS tags for a Spanish treebank. This improved the performance of a lexicalized PCFG parser by several percent. While generative constituency parsing and discriminative dependency parsing are different approaches to the same problem, these results provide further support for the potential of morphological information to enhance statistical parsing performance.

3.3.3 Agreement Modeling in Other Tasks

Although morphological agreement has not been exploited fully in parsing, it has appeared in a variety of other NLP tasks. Lee et al. (2011) modeled agreement in Arabic by looking for words with matching gender- and number-marking suffixes, thereby increasing the F1-score of their morphological analyzer by 4.6%. Minkov et al. (2007) used agreement features based on morphological annotations to predict target word forms when translating from English to Russian and Arabic, for example, triggering an agreement feature for a potential target word which shares the same value for a

morphological attribute as the previously predicted target word. This increased word form prediction accuracy by several percentage points. Rajkumar and White (2010) created a set of English-specific features which capture agreement for number between nouns and verbs, and for animacy between nouns and wh-pronouns. These features are used for sentence ranking in CCG realization, and result in significant reduction of both number and animacy agreement errors.

3.3.4 Summary

The dependency parsing systems which utilize morphological information fall roughly into several groups. Many of them, including most of the CoNLL shared task participants, are language-independent. These systems generate machine learning features to capture morphological information in one of several ways: by using the list of a token's attributes as an atomic feature; by using each element of the list separately; or by using each pair from the cross product of the token's list with the list of a potential head. These features are then fed into a parser. The creators of many of these systems experimented with feature selection – including different subsets of the available morphological attributes to determine the optimal combination (for example, including case and gender, but not number). This process is language-independent in the sense that it can be done for any language, and could potentially be automated, but language-specific in that the results cannot be transferred to other languages.

Another set of systems takes language-specific approaches, such as using handcrafted rules to generate finer POS tags from morphological information, or generating different features for tokens with different POS tags. Although most of these approaches could be realized for other languages with comparable morphological properties, this would require creating models on a per-language basis.

Nearly all of these systems learn agreement relationships between morphological attributes statistically. Only two systems modeled agreement explicitly, generating specific features when agreement between tokens is present, and neither of these

did so cross-linguistically. Yet, from the results mentioned above, it is clear that morphological information is useful for parsing, and even if agreement is not modeled explicitly, the increases in performance due to incorporating such information are presumably due at least in part to the presence of agreement relationships.

Chapter 4
METHODOLOGY

Following are details of the agreement modeling system, data collection, and experimental setup.

4.1 Implementation of Agreement Model

4.1.1 Motivation and Theory

Many parsers rely on word order to establish dependencies, so they generally perform better on languages with less flexible word order (Nivre et al., 2007). Since languages with more flexible word order tend to display more inflection, parsers should be able to leverage this information to compensate for the variation in word order. In essence, they could rely on word order in languages where it is inflexible (and which generally exhibit less morphology), and on morphology in languages where it is present (and where word order is generally more flexible).

Tesnière, the father of dependency grammar, suggested this, stating that agreement plays a fundamental role in establishing dependencies which do not appear in a linear sequence (1959). Knowledge of whether two words agree or disagree, combined with their POS tags, can help a parser to determine whether they are linked. For instance, it might learn that a determiner marked for genitive case is probably headed by a similarly marked noun. If agreement is modeled explicitly – abstracting away from specific morphological attributes and instead encoding agreement between tokens – such a parser need only learn that a determiner and noun generally agree for case, rather than learning this correspondence separately for each possible case.

This modeling functions as a type of backoff, allowing the parser to extract more

general information from morphological marking. Abstracting away from the actual values makes the data less sparse, since the learner is only cataloging occurrences of agreement, rather than occurrences of agreement for a specific value. This should lead to higher parsing accuracy, especially when training on fewer sentences.

As described earlier, agreement has been modeled this way in limited and language-specific contexts. But language-independent approaches thus far (e.g., MSTParser and MaltParser) have generally used morphological information in a linguistically naïve way, listing the morphological attributes of each token without taking advantage of patterns in these attributes which might be helpful. In contrast, I take a linguistically informed approach – while still maintaining language independence – by explicitly modeling agreement between head and dependent morphology.

4.1.2 Agreement Model Features

The agreement model consists of a small set of features which encapsulate agreement or disagreement between two tokens for an attribute which appears on both ("symmetric" features). In the case of an attribute appearing on one token but not the other, a third type of feature is triggered ("asymmetric"). In order to test the agreement model, I added code to MSTParser to generate the agreement features. (Source code for the added features can be found in Appendix A.)

Since MSTParser breaks down every parse into a set of arcs, agreement features are defined at the arc level. Each arc is defined as a head and dependent pair, and each of those tokens has a list of morphological features in the normalized form `attribute=value`. The head and dependent lists are compared, and for every attribute which is present in both, either an agreement or a disagreement feature is added, depending on whether the head and dependent have the same value for that attribute. These symmetric features encapsulate the agreeing attribute, but not its value, as well as the coarse POS tags of the head and the dependent. If an attribute is present in only one of the lists, a feature is added encoding whether the token is the head or

Table 4.1: Agreement model feature templates. `attr` is the morphological attribute (e.g., `case`); `value` is the value for that attribute (e.g., `ACC`). `label` is the dependency arc label (e.g., `SBJ`). {`head|dependent`} is `head` when the attribute is present on the head but not the dependent, and `dependent` in the opposite case. `hdPOS` and `dpPOS` are the coarse POS tags of the head and dependent. Each line represents one feature.

Agreement features
<attr>_agrees,head=<hdPOS>,dependent=<dpPOS>
<attr>_agrees&label=<label>,head=<hdPOS>,dependent=<dpPOS>
Disagreement features
<attr>_disagrees,head=<hdPOS>,dependent=<dpPOS>
<attr>_disagrees&label=<label>,head=<hdPOS>,dependent=<dpPOS>
Asymmetric features
{head|dependent}_<attr>=<value>,head=<hdPOS>,dependent=<dpPOS>
{head|dependent}_<attr>=<value>&label=<label>,head=<hdPOS>,dependent=<dpPOS>

the dependent, the single morphological feature (attribute and value), and the two POS tags. A version of each of these features conjoined with the dependency label is added as well. Table 4.1 presents a summary of these features.

This approach discards some of the information which is retained by the original feature set. In the case of agreement or disagreement between two tokens, it encodes the relevant information (agreement or disagreement) but discards the value of the attribute, which should not affect the likelihood of the dependency. In the case of an attribute marked on only one token, the attribute and value are retained. Furthermore, no conjunctions of dissimilar attributes (e.g., features including the case marked on one token and the gender marked on another) are retained.

4.2 Data Collection and Preparation

The modified MSTParser was tested on a range of dependency treebanks with varying amounts of morphological annotation. They are listed in Table 4.2.

For this research, I adopted the treebank format used for the CoNLL shared tasks. This format includes one token on each line, along with syntactic information

Table 4.2: Language, ISO 639-2 code, treebank name, total number of sentences, reference size, average number of morphological attributes per token, and reference for each treebank used, ordered by average number of attributes.

Language	ISO	Treebank	Num. sents.	Ref. size	Avg. atts.	Reference
Hindi-Urdu	hin	HUTB	3,855	2,800	3.6	Bhatt et al., 2009
Hungarian	hun	Szeged DTB	92,176	9,000	3.3	Vincze et al., 2010
Czech	ces	PDT 1.0	73,068	9,000	2.8	Hajič, 1998
Tamil	tam	TamilTB v0.1	600	600	2.8	Ramasamy & Žabokrtský, 2011
Slovene	slv	SDT	1,998	1,500	2.6	Džeroski et al., 2006
Danish	dan	DDT	5,512	5,500	2.4	Kromann, 2003
Basque	eus	3LB*	3,175	2,800	2.4	Aduriz et al., 2003
Dutch	nld	Alpino	13,735	9,000	2.4	Van der Beek et al., 2002
Latin	lat	LDT	3,423	2,800	2.4	Bamman & Crane, 2006
Bulgarian	bul	BulTreeBank	13,221	9,000	2.1	Simov et al., 2004
Greek (ancient)	grc	AGDT	21,104	9,000	2.1	Bamman et al., 2009
Finnish	fin	Turku	4,307	2,800	2.0	Haverinen et al., 2010
German	deu	NEGRA	3,427	2,800	2.0	Brants et al., 1999
Turkish	tur	METU-Sabanci	5,620	5,500	1.6	Oflazer et al., 2003
Catalan	cat	CESS-ECE*	3,512	2,800	1.5	Martı et al., 2007
Arabic	ara	PADT 1.0	2,367	2,300	1.2	Hajic et al., 2004
Italian	ita	TUT	2,858	2,800	1.1	Bosco et al., 2000
Portuguese	por	Floresta	9,359	9,000	1.0	Afonso et al., 2002
Hebrew (modern)	heb	DepTB	6,214	5,500	0.9	Goldberg, 2011
English	eng	Penn*	49,208	9,000	0.4	Marcus et al., 1993
Chinese	cmn	Penn Chinese	28,035	9,000	0.0	Xue et al., 2005

*Acquired as part of NLTK (Bird et al., 2009)

detailed in Table 4.3. (The `PHEAD` and `PDEPREL` fields provide space to include both projective and non-projective parses in the same file; they are optional, and ignored by MSTParser.) There is assumed to be a root token external to each sentence with `ID=0`.

Of the twenty-one treebanks I gathered, all but six – Hindi-Urdu, Latin, Greek, German, Arabic, and Chinese – were already in the CoNLL format. Of these, the Hindi-Urdu, Latin, Greek, and German treebanks were in different dependency formats which were easily converted to CoNLL. The Arabic data was in the FS ("fea-

Table 4.3: Information included in CoNLL treebank format.

Field	Description
ID	position in sentence
FORM	surface form of token
LEMMA	lemma
CPOS	coarse part-of-speech tag
FPOS	fine part-of-speech tag
MORPH	list of morphological attributes
HEAD	ID of head
DEPREL	dependency label
PHEAD	ID of projective head (not used)
PDEPREL	label of projective dependency (not used)

ture structure") format,[1] which was converted to the CoNLL format by means of the any2any[2] and padt2tab[3] utilities.

I generated the Chinese data by using the Penn2Malt converter[4] to convert the Penn Chinese Treebank constituency parses to dependency parses, and a script to convert the resulting Malt-TAB data to CoNLL format.

Not all of the treebanks contained values for all the fields provided in the format. Every treebank included values for ID, FORM, HEAD, DEPREL, and at least one POS tag field. The lemma and morphological attributes were not present in all of the datasets. Since the goal was not to make comparisons between performance on multiple languages, but rather to assess the effects of agreement modeling on each language separately, I made no attempt to ensure that comparable amounts or types of data were used across languages.

All except the Bulgarian, Chinese, Danish, English, German, and Hebrew datasets contained lemmas. The Arabic dataset contained lemmas for only some tokens.

---

[1] http://ufal.mff.cuni.cz/pdt/Corpora/PDT_1.0/Doc/fs.html

[2] http://ufal.mff.cuni.cz/pdt/Utilities/cstsfs/index.html

[3] http://ilk.uvt.nl/conll/software.html#other

[4] http://w3.msi.vxu.se/~nivre/research/Penn2Malt.html

The POS tagsets used in the treebanks varied widely. The Basque, Bulgarian, Catalan, Czech, Danish, Dutch, Portuguese, Slovene, Tamil, and Turkish data included two POS tags for each word: one coarse and one fine tag. The others contained only one tag per word. I normalized the coarse POS tags in all treebanks to a universal tagset for several reasons: in order to ensure that every treebank had coarse tags for use in the agreement features, to make the features easier to interpret, and to allow for cross-linguistic generalization based on POS tag in future work. This was done using the set of twelve tags suggested by Petrov et al. (2011), who also provide original-to-universal tag mappings for a number of treebanks. For the others, I generated mappings based on treebank documentation. These appear in Appendix B.

The format and quantity of morphological annotations present in the data also varied from one treebank to another. (Table 4.2 lists the average number of attributes per token for each treebank.) The Hindi-Urdu treebank had the most morphological information, averaging 3.6 attributes per token. Two treebanks with no morphological information at all, the Penn Chinese and English treebanks, were included. For the English data, I generated morphological annotations based on the original POS tags, consisting primarily of person and number information for nouns and verbs, and person, number, and case information for pronouns; the mapping I used can be found in Appendix C.

The German NEGRA Corpus includes detailed morphological annotations for about 3,400 sentences (out of 20,600 total). Only that part of the corpus was used for this work.

Note that the amount of morphological information present in a treebank is a function of both the morphological properties of the language and the annotation guidelines. That is, the annotation for a specific token does not necessarily encode all of the morphological information actually present in the token. Accordingly, the average number of attributes per token cannot be taken as a measure of the morphological complexity of a language. Furthermore, the presence of a morphological

attribute does not imply that a token participates in an agreement relationship; it merely encodes some piece of morphological information about the token. Finally, annotation guidelines vary as to whether they provide for the explicit marking of morphological properties which are inherent to a token (e.g., gender on nouns), not marked by separate affixes.

I normalized the format of all morphological attributes to the form `attribute=value`, where `attribute` is a morphological attribute, such as case, person, or gender, and `value` is a possible value, such as nominative, second person, or feminine. For treebanks that provide values only (rather than attribute-value pairs), this normalization included the addition of attribute names derived from the annotation guidelines. Person, number, gender, and case attributes appeared often; also included in some data were verb tense, adjective degree, and pronoun type (e.g., personal, possessive, or reflexive). All attributes present in the data were normalized, regardless of whether or not they participate in any agreement relationships. Details of the information included in each treebank appear in Appendix D.

A sample Czech sentence in IGT and the CoNLL treebank format, with normalized POS tags and morphological information, along with the agreement features it would trigger, is shown in Table 4.4.

Many treebanks included data from multiple domains. In order to minimize the consequences of this when parsing different amounts of data, the order of sentences in each treebank was randomized. All punctuation and other tokens were retained. No sentences were filtered or removed except in extremely rare cases of problematic or inconsistent annotations; for example, if an annotation error caused a dependency cycle within a sentence (i.e., a token was annotated as a descendant of itself), no effort was made to correct the error, and the sentence was removed. This resulted in the removal of no more than two sentences per treebank.

Table 4.4: Sample sentence in IGT and CoNLL format, and agreement features generated.

Vznikají	zbytečné	konflikty.
arise.3RD.PL	unnecessary.PL.INAN.NOM	conflicts.PL.INAN.NOM
'Unnecessary conflicts arise.'		

ID	FORM	LEMMA	CPOS	FPOS	MORPH	HEAD	DEPREL
1	Vznikají	vznikat	VERB	B	num=PL|per=3	0	ROOT
2	zbytečné	zbytečný	ADJ	A	num=PL|gen=I|case=NOM	3	ATR
3	konflikty	konflikt	NOUN	N	num=PL|gen=I|case=NOM	1	SBJ

Arc from 2 to 3	num_agrees,head=NOUN,dependent=ADJ num_agrees&label=ATR,head=NOUN,dependent=ADJ gen_agrees,head=NOUN,dependent=ADJ gen_agrees&label=ATR,head=NOUN,dependent=ADJ case_agrees,head=NOUN,dependent=ADJ case_agrees&label=ATR,head=NOUN,dependent=ADJ
Arc from 3 to 1	num_agrees,head=VERB,dependent=NOUN num_agrees&label=SBJ,head=VERB,dependent=NOUN head_per=3RD,head=VERB,dependent=NOUN head_per=3RD&label=SBJ,head=VERB,dependent=NOUN dependent_gen=I,head=VERB,dependent=NOUN dependent_gen=I&label=SBJ,head=VERB,dependent=NOUN dependent_case=NOM,head=VERB,dependent=NOUN dependent_case=NOM&label=SBJ,head=VERB,dependent=NOUN

4.3 Experimental Setup

I downloaded MSTParser 0.4.3c[5] and built it on a Linux system. It was run in second-order mode for all experiments.

All experiments were performed using 5-fold cross-validation, assigning the first sentence (after randomization) to the first fold, the second sentence to the second fold, etc. Reported accuracies, run times, and feature counts are averages over all five folds.

The feature configurations used were the original MSTParser morphological fea-

[5] http://sourceforge.net/projects/mstparser/

ture set (`orig`), the agreement modeling feature set (`agr`), the union of both sets (`agr+orig`), and neither set (`no-morph`). All other MSTParser features (i.e., those representing word and POS sequences) were left untouched and included in all experiments.

Two modifications to the parser, aside from the addition of the agreement features, were made. The reading of treebank data was modified slightly in order to ensure that all feature sets performed the same on data containing no morphological annotations, such as the Chinese treebank. Depending on the treebank, this slightly increased or decreased the performance of the `orig` system by less than 0.5%. With this modification, all feature configurations performed identically on treebanks which do not include morphological information, providing a consistent baseline. Details of this modification appear in Appendix E. The second modification involved adding a non-morphological feature to the parser, and is described in section 5.4 and Appendix G.

The collected treebanks range in size from 600 to around 92,000 sentences. I ran experiments for each treebank on multiple dataset sizes in order to assess the performance of agreement modeling with varying amounts of training data. For each treebank, I designated a "reference size" on which to report results; this was 9,000 sentences, or the largest dataset size smaller than the size of the treebank for treebanks of less than 9,000 sentences. (The reference size for each treebank is listed in Table 4.2.)

MSTParser provides an evaluation module which calculates unlabeled accuracy (percentage of tokens with correctly assigned heads, ignoring arc labels) and labeled accuracy (percentage of tokens with correctly assigned heads and labels) as well as unlabeled and labeled complete correct (percentage of sentences which were correct). I focused on the unlabeled accuracy scores; the labeled accuracy scores displayed nearly identical trends to the unlabeled scores.

Chapter 5
EXPERIMENTS AND RESULTS

I first ran the system twice on each treebank with the `no-morph` feature configuration: once in projective mode, and once in non-projective. For each treebank, the mode which performed better across all dataset sizes was used for all subsequent experimentation on that treebank. When projective and non-projective modes worked equally well, non-projective was chosen. Based on this, only the Bulgarian, Catalan, Italian, Hebrew, English, and Chinese treebanks were parsed projectively.

I then ran the parser with each of four morphological feature configurations (`orig`, `agr`, `agr+orig`, `no-morph`) on each treebank.

5.1 Overall Results

In general, using the agreement features increased accuracy on all languages and at all dataset sizes. The magnitude of this increase varied from one treebank to another. Using only the agreement features (`agr`) was generally better than using the agreement features in combination with the original features (`agr+orig`), presumably because the sheer quantity of features generated by the original feature set overwhelmed the machine learning algorithm. For this same reason, performance was noticeably faster with `agr` than with `orig` and `agr+orig`.

Table 5.1 displays the unlabeled accuracy, runtime, and number of machine learning features when parsing each treebank using all four feature configurations at the reference size. The highest accuracy for each language is highlighted. Run time and number of features for `orig`, `agr`, and `agr+orig` are given as percent change relative to `no-morph`; raw data is given in Appendix F.

Table 5.1: Reference size, unlabeled accuracy, run time in seconds, and feature count in millions, for all treebanks containing morphological information. Run time and number of features for `orig`, `agr`, and `agr+orig` are given as percent change relative to `no-morph`. Languages appear in decreasing order of average number of morphological attributes per token.

	`no-morph`			`orig`			`agr`			`agr+orig`		
Lang	UAC	time	feats	UAC	Δtime	Δfeats	UAC	Δtime	Δfeats	UAC	Δtime	Δfeats
hin	90.0	1.4k	1.6	92.0	116%	893%	**93.8**	50%	1%	93.0	144%	893%
hun	87.9	4.6k	5.3	88.7	201%	687%	**90.3**	10%	0%	89.9	159%	687%
ces	80.9	3.3k	4.8	81.6	71%	454%	**85.5**	27%	0%	84.5	114%	454%
tam	79.0	0.1k	0.5	79.7	237%	329%	**82.1**	64%	1%	81.1	279%	330%
slv	80.8	0.8k	1.0	80.4	102%	352%	**81.8**	21%	0%	80.8	129%	353%
dan	87.8	2.0k	1.6	88.4	71%	256%	**89.3**	24%	0%	**89.3**	86%	256%
lat	61.7	1.8k	1.6	65.0	54%	306%	**70.3**	91%	0%	68.6	119%	306%
nld	88.2	2.0k	3.6	89.0	83%	270%	**90.5**	16%	0%	90.3	98%	270%
eus	78.7	0.7k	1.7	80.2	80%	229%	**82.3**	10%	0%	**82.3**	78%	230%
bul	89.9	1.7k	2.6	90.2	60%	221%	**93.0**	14%	0%	92.5	54%	222%
grc	74.9	8.6k	3.8	77.0	36%	314%	**80.7**	45%	0%	79.5	70%	314%
deu	90.0	0.9k	1.3	90.8	33%	189%	**92.0**	1%	0%	91.7	50%	186%
fin	73.3	0.7k	2.4	76.3	74%	244%	**79.1**	23%	0%	78.7	84%	245%
tur	80.2	1.2k	2.1	81.5	13%	178%	81.6	–2%	0%	**81.7**	29%	178%
cat	81.8	3.0k	2.5	81.9	1%	142%	**84.9**	–9%	0%	84.0	–2%	143%
ara	77.6	5.4k	1.8	77.7	20%	100%	**78.2**	–8%	0%	78.0	4%	100%
ita	88.4	4.2k	1.8	88.9	–2%	59%	90.2	9%	0%	**90.3**	6%	59%
por	88.1	6.4k	5.0	88.2	18%	46%	**89.0**	–3%	0%	88.9	27%	46%
heb	87.4	4.3k	3.1	87.4	–18%	31%	**89.2**	–16%	0%	89.1	–5%	31%
eng	88.1	5.2k	3.1	88.0	5%	7%	**90.6**	3%	0%	**90.6**	–9%	7%
cmn	**82.4**	7.5k	6.0	**82.4**	37%	0%	**82.4**	16%	0%	**82.4**	23%	0%

In most cases, `agr` was the best configuration, outperforming `orig` by margins of up to 5.3% (Latin). On two treebanks (Turkish and Italian), `agr+orig` outperformed `agr` slightly, and on three others, exclusive of Chinese (Danish, Basque, and English), those configurations performed the same. However, `agr` always outperformed `orig` (except on Chinese, where results were identical for all configurations, as expected).

In all cases, the number of features is considerably higher for the two feature configurations including the original feature set. This is because that set includes exhaustively features based on all possible combinations of morphological attributes on parent and child tokens, generating a large number of uninformative features. The agreement feature set is very small compared to the number of non-morphological features, so feature counts for `agr` and `no-morph` are similar.

Run time data is noisy, due to the variety of cluster nodes (with varying processor, memory, and disk configurations) on which the parser was run. Nonetheless, it is clear that parsing jobs with high feature counts had similarly high run times.

Figure 5.1 presents the increases in performance due to various feature configurations graphically; Figure 5.2 presents this same information as the error reduction of `orig`, `agr`, and `agr+orig` relative to `no-morph`. In both, languages are listed in order of average morphological attributes per token, from the most attributes (Hindi) to the least (Chinese).

Despite its relative lack of morphological inflection, English shows a fairly high error reduction; this is because parsing performance on English was already very high. Similarly, error reduction on some of the morphologically rich languages is lower because, although absolute improvement was high, baseline performance on those languages was low. As mentioned earlier, Chinese has no morphological agreement, so all of the feature configurations performed the same. It is included here only as a representative of the class of analytic languages.

Calculating the correlation coefficient (Pearson's r) between average number of morphological attributes per token and error reduction relative to `no-morph` gives $r = 0.608$ for `orig`, $r = 0.560$ for `agr`, and $r = 0.428$ for `agr+orig`, with $p < 0.01$ for the first two and $p < 0.10$ for the last. This indicates moderate correlations for all feature sets.

The strength of the correlations seen here is dependent on several factors. Languages differ in what information is marked morphologically, and in the quantity of agreement relationships involving this information. Similarly, annotation schemes vary in what morphological information they encode, and in how important that information is to agreement specifically; it would be possible to annotate a great deal of information for each token, none of which is relevant to discovering dependencies via agreement, or to omit annotations of inherent attributes which would be useful. Some morphologically complex languages have rigid word order, leading to better perfor-

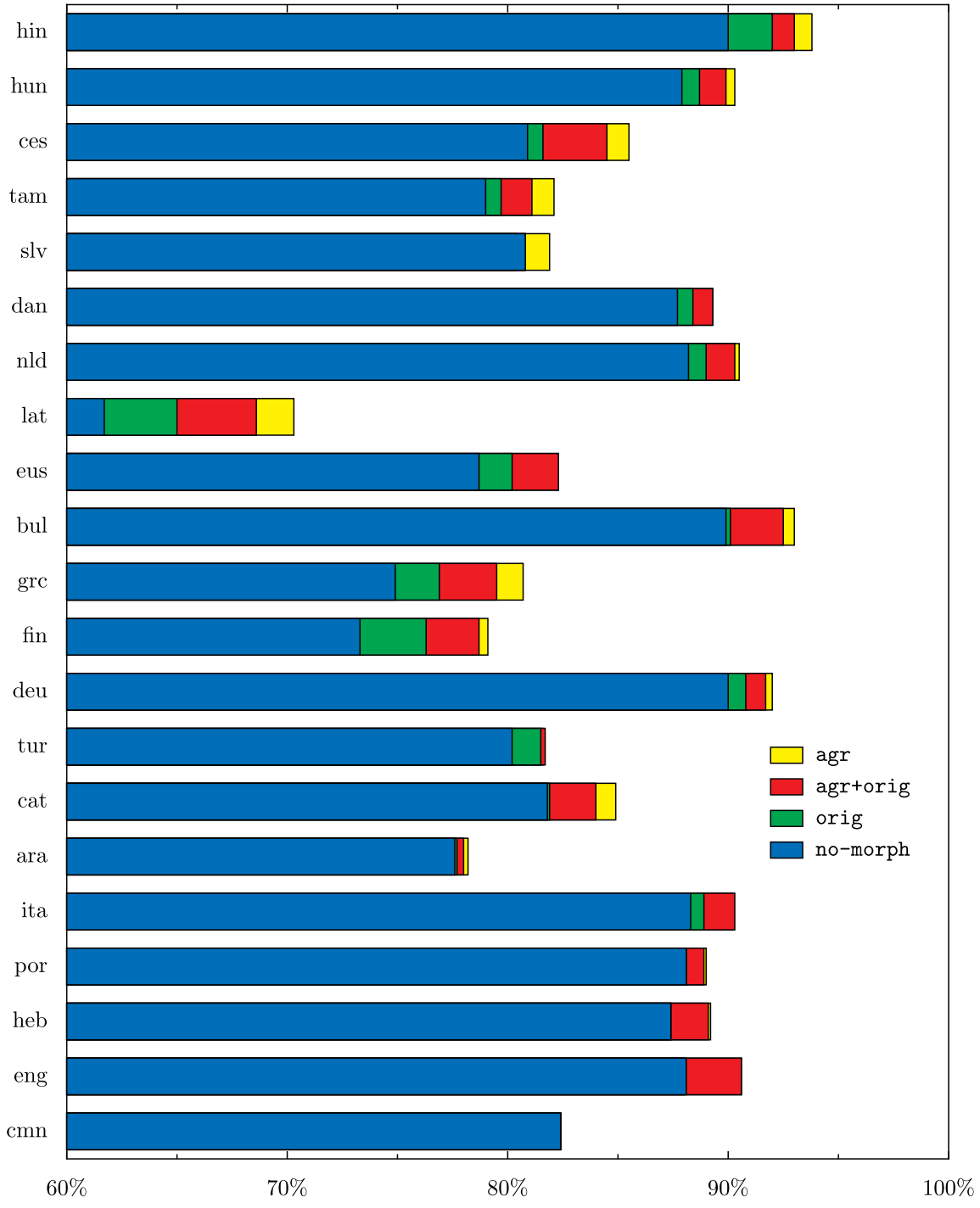


Figure 5.1: Unlabeled accuracy of feature configurations by treebank. Note that, in this chart, each colored bar essentially hides those preceding it in the order listed; this means that some information is lost if a configuration performs better than one before it, e.g., if `no-morph` outperforms `orig`.

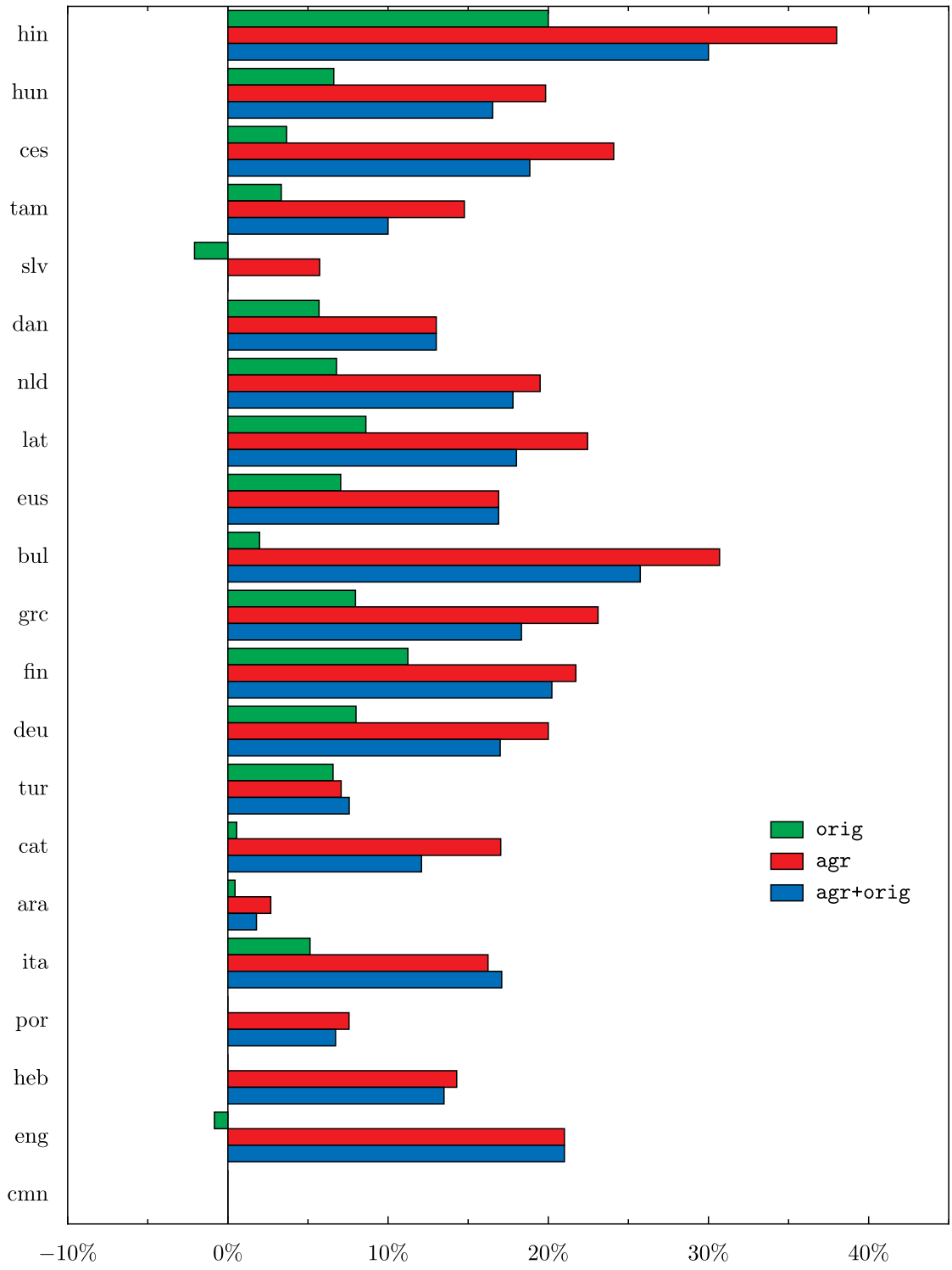
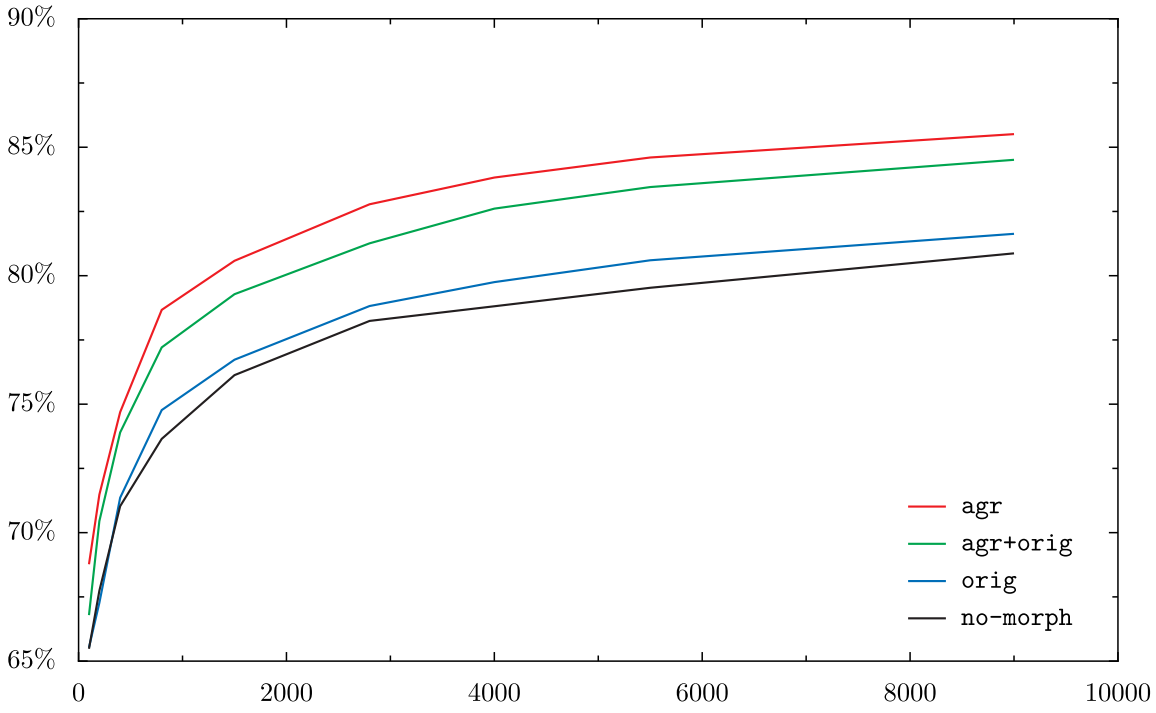


Figure 5.2: Error reduction of feature configurations (relative to `no-morph`) by treebank.



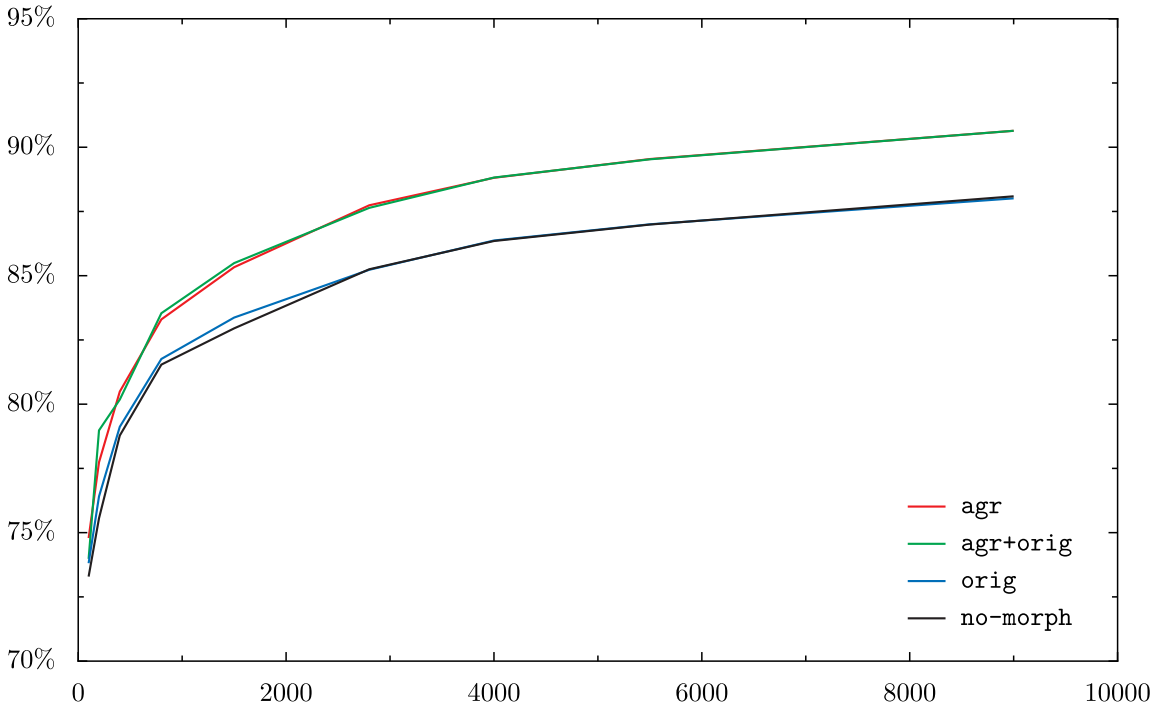
	100	200	400	800	1500	2800	4000	5500	9000
`no-morph`	65.5	67.7	71.0	73.7	76.1	78.2	78.8	79.5	80.9
`orig`	65.5	67.3	71.4	74.8	76.7	78.8	79.8	80.6	81.6
`agr`	68.8	71.5	74.7	78.7	80.6	82.8	83.8	84.6	85.5
`agr+orig`	66.8	70.5	73.9	77.2	79.3	81.3	82.6	83.5	84.5

Figure 5.3: Unlabeled accuracy vs. dataset size in sentences, Czech treebank.

mance with no morphological features at all, and limiting the amount of improvement that is possible. Finally, it is possible that a stronger correlation is obscured by other effects due to feature set design: as we will find in section 5.4, the `agr` features capture additional information unrelated to morphology.

5.2 Improvement vs. Dataset Size

Figures 5.3 and 5.4 present unlabeled accuracy as a function of dataset size when parsing Czech and English with all four feature configurations. Improvement with `agr` is roughly uniform across all dataset sizes except the smallest few, where it



	100	200	400	800	1500	2800	4000	5500	9000
`no-morph`	73.3	75.6	78.8	81.5	83.0	85.3	86.4	87.0	88.1
`orig`	73.8	76.4	79.1	81.8	83.4	85.2	86.4	87.0	88.0
`agr`	74.8	77.7	80.5	83.3	85.3	87.7	88.8	89.5	90.6
`agr+orig`	74.0	79.0	80.2	83.5	85.5	87.6	88.8	89.5	90.6

Figure 5.4: Unlabeled accuracy vs. dataset size in sentences, English treebank.

was slightly lower. This was the trend seen for all languages, and contradicted the hypothesis that the agreement features would be more helpful when less data was available. It is possible that the improvement would decrease on datasets of more than 9,000 sentences, although the results for sizes up to 9,000 do not suggest this.

For Czech, the absolute improvement for `agr` over `orig` averaged around 4%. Parsing with `agr+orig` provided less performance improvement (around 2.5% on average), presumably because the great number of `orig` features (about 21.8 million at 9,000 sentences) overwhelmed the `agr` features (about 10,000). For English, the performance of `agr` and `agr+orig` was nearly identical; this is expected, because the

English data has fewer attributes per token, which means fewer `orig` features are generated (about 235,000 at 9,000 sentences).

5.3 Gold vs. Automatically Predicted Annotations

The Hebrew treebank used for this research includes both automatically generated and gold standard (oracle) POS and morphological annotations.[1] In order to test how sensitive the agreement features are to automatically predicted morphological information, tests were run on both versions at the reference size.

These results are not directly comparable to those of Goldberg and Elhadad (2009), because of the parser modifications, POS tag normalization, and use of cross-validation described earlier. Comparing results qualitatively, I find less sensitivity to the automatic tags overall, and that the `orig` features improve accuracy even when using automatic tags.

Results of this experiment appear in Table 5.2. Using the automatically tagged data affects all feature sets negatively by 2.1% to 2.9%. Since the `no-morph` parser was affected the most, it appears that most of this decrease is due to errors in the POS tags, rather than to the morphological annotations. The `orig` features compensate for this slightly (0.2%), and the `agr` features far more (0.8%); this demonstrates that including even automatic morphological information can compensate for incorrect POS tags, and that the `agr` feature configuration is the most robust, showing the least sensitivity to the automatic tags.

5.4 PPL feature

During training, the machine learning algorithm integrated into the parser learns a weight for every feature which is generated. These weights are used during parsing to score each possible parse of a sentence; the score of a potential parse is the sum of

[1] It also includes a version with automatically tokenized data, which was not used in these trials.

Table 5.2: Unlabeled accuracy of four feature configurations on the 5500-sentence Hebrew dataset, with gold POS and morphological annotations and with automatic annotations.

Feature configuration	Accuracy on gold data	Accuracy on automatic data	Difference
`no-morph`	87.4	84.5	−2.9
`orig`	87.4	84.7	−2.7
`agr`	89.3	87.2	−2.1
`agr+orig`	89.1	86.9	−2.2

the weights of all the features it triggers. Thus, features with high weights often have the most impact on scores.

Examining the feature weights from the first cross-validation fold when running the `agr` feature configuration on the 9,000-sentence Czech dataset indicated that 323 of the 1,000 highest-weighted features are agreement features. Of these, 79 are symmetric ("agrees" or "disagrees") `agr` features, and 244 are asymmetric. Furthermore, of the 20 highest-weighted features, 14 are labeled asymmetric `agr` features; the highest-weighted symmetric `agr` feature comes in at number 19. This was unexpected, as the symmetric features would seem to be more valuable to a parser.

This suggested that the labeled asymmetric `agr` features might be important for reasons other than their modeling of morphological information. Looking carefully at the MSTParser feature set revealed that it does not include a feature which incorporates head POS, dependent POS, and dependency label. I hypothesized that the labeled asymmetric `agr` features were filling this gap, since they include these three items, in addition to either the head or dependent morphological attribute.

In order to test this, I added a single feature template to MSTParser which encapsulates head POS, dependent POS, and dependency label (the POS-POS-label, or PPL, feature); Appendix G contains details of this modification. Since this template does not include morphological information, it is more general than the labeled asymmetric `agr` features, and should therefore appear more often. Running a subsequent

Table 5.3: Number of feature templates of each type appearing in the 1,000 highest-weighted features, with and without the PPL feature added to the system. This data comes from the first cross-validation fold, parsing the 9,000-sentence Czech dataset.

Feature type	**Count with `agr` features only**	**Count with `agr` and PPL features**
symmetric "agrees"	51	38
symmetric "disagrees"	28	11
asymmetric	244	187
PPL	–	278

experiment on the Czech data and looking at feature weights from the same cross-validation fold, 278 of the 1,000 highest-weighted features, and 13 of the 20 highest-weighted, were PPL features. This indicated that the improvement seen with `agr` features was indeed due partly to their inclusion of features combining head POS, dependent POS, and dependency label. Table 5.3 includes summaries of highly-weighted features with and without the PPL feature.

All four feature configurations were run on the full set of treebanks with the PPL feature incorporated into MSTParser. Results of this experiment appear in Table 5.4, Figure 5.5, and Figure 5.6, analogous to the results presented earlier. Raw results appear in Appendix F. With the inclusion of the PPL feature, the performance increases from `orig` to `agr` are generally smaller, with a maximum of 4.6%. This is seen especially on languages with less morphological information, such as English and Hebrew; this indicates that for those languages, most of the previous improvement was due not to agreement modeling, but to the lack of a PPL feature in the original MSTParser.

With the PPL feature included, `agr+orig` outperformed `agr` on only one treebank (Basque), but matched it on five treebanks excluding Chinese (Dutch, Arabic, Italian, Portuguese, and English) – a larger set than was seen earlier.

Calculating Pearson's r based on the new error reduction data gives a stronger

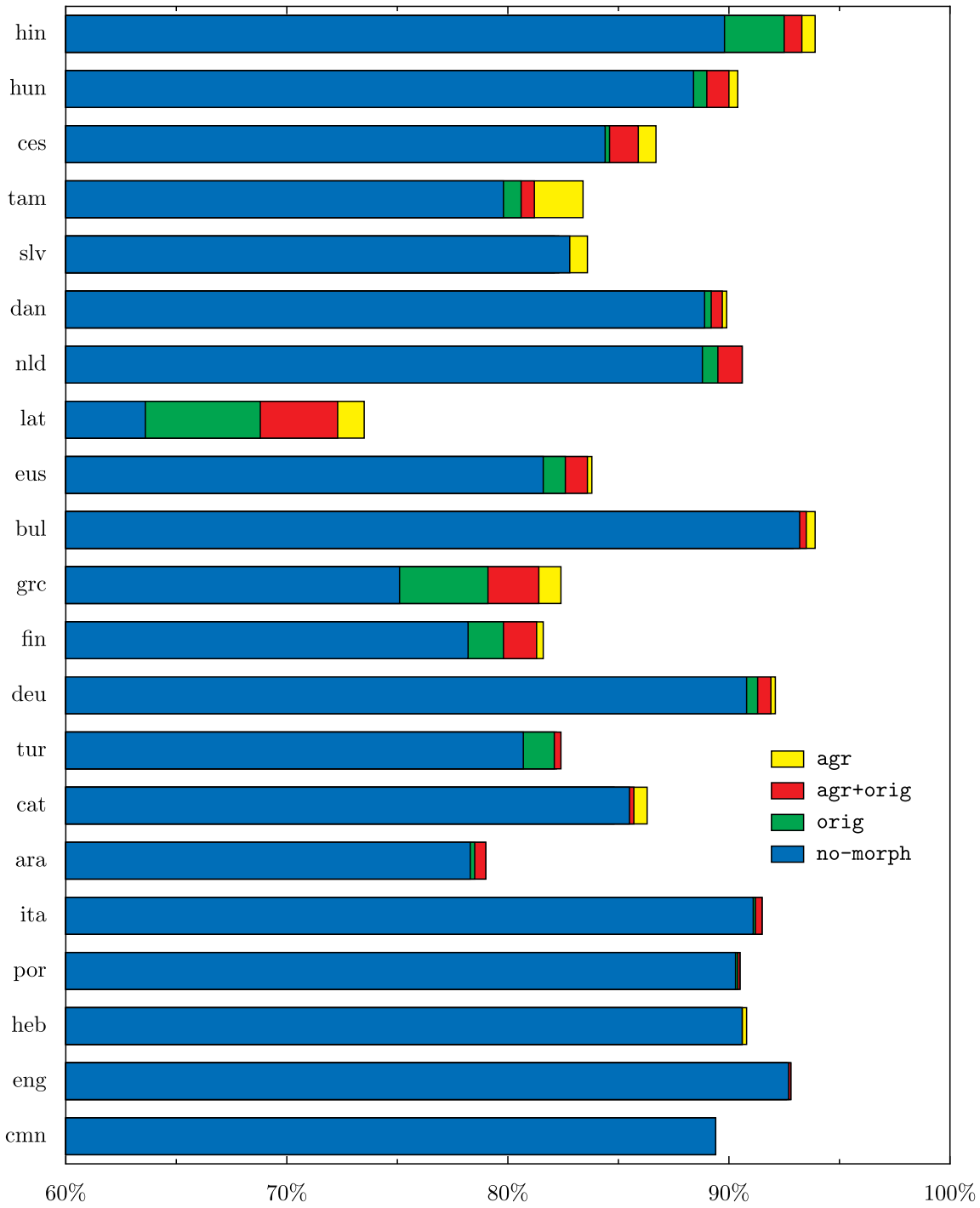


Figure 5.5: Unlabeled accuracy of feature configurations by treebank, with PPL feature included. Note that, in this chart, each colored bar essentially hides those preceding it in the order listed; this means that some information is lost if a configuration performs better than one before it, e.g., if `no-morph` outperforms `orig`.

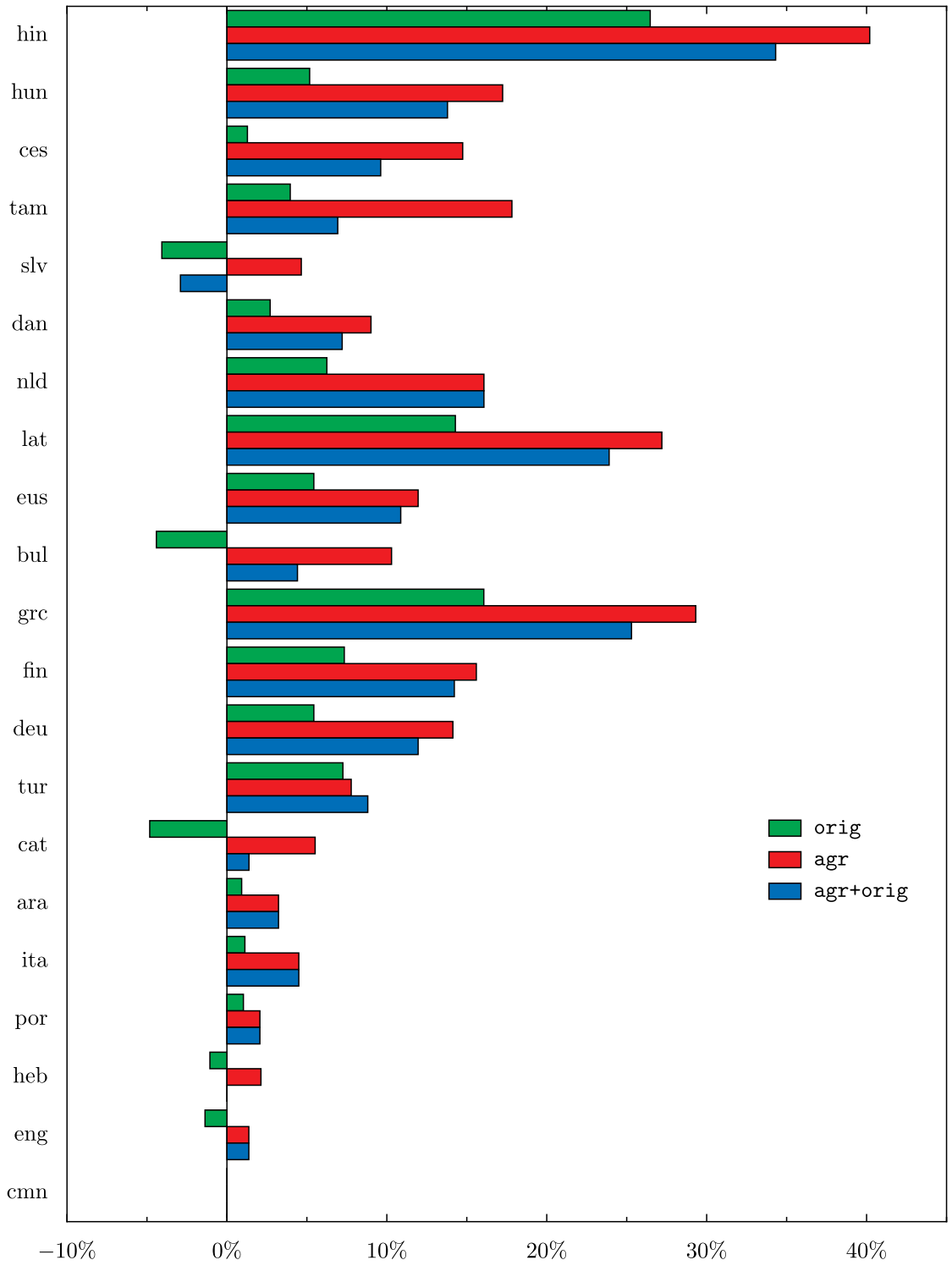


Figure 5.6: Error reduction of feature configurations (relative to `no-morph`) by treebank, with PPL feature included.

Table 5.4: Reference size, unlabeled accuracy, run time in seconds, and feature count in millions, for all treebanks containing morphological information, with PPL feature included. Run time and number of features for `orig`, `agr`, and `agr+orig` are given as percent change relative to `no-morph`. Languages appear in decreasing order of average number of morphological attributes per token.

	`no-morph`			`orig`			`agr`			`agr+orig`		
Lang	UAC	time	feats	UAC	Δtime	Δfeats	UAC	Δtime	Δfeats	UAC	Δtime	Δfeats
hin	89.8	1.4k	1.6	92.5	116%	892%	**93.9**	62%	1%	93.3	169%	893%
hun	88.4	4.4k	5.3	89.0	175%	687%	**90.4**	15%	0%	90.0	199%	687%
ces	84.4	3.4k	4.8	84.6	89%	453%	**86.7**	8%	0%	85.9	107%	454%
tam	79.8	0.3k	0.5	80.6	61%	329%	**83.4**	0%	1%	81.2	94%	329%
slv	82.8	0.6k	1.0	82.1	99%	352%	**83.6**	31%	0%	82.3	142%	352%
dan	88.9	2.4k	1.6	89.2	35%	256%	**90.0**	2%	0%	89.7	54%	256%
lat	63.6	2.2k	1.6	68.8	21%	306%	**73.4**	4%	0%	72.3	30%	306%
nld	88.8	1.9k	3.6	89.5	89%	270%	**90.6**	23%	0%	**90.6**	107%	270%
eus	81.6	0.7k	1.7	82.6	70%	230%	**83.8**	20%	1%	83.6	77%	231%
bul	93.2	1.9k	2.6	93.0	39%	221%	**93.9**	–3%	0%	93.5	52%	221%
grc	75.1	9.9k	3.8	79.1	20%	313%	**82.4**	34%	0%	81.4	44%	314%
deu	90.8	0.9k	1.3	91.3	41%	186%	**92.2**	7%	0%	91.9	50%	186%
fin	78.2	0.8k	2.4	79.8	51%	244%	**81.6**	8%	0%	81.3	68%	245%
tur	80.7	1.0k	2.1	82.1	84%	178%	82.2	41%	0%	**82.4**	45%	178%
cat	85.5	2.5k	2.5	84.8	19%	142%	**86.3**	–12%	0%	85.6	54%	142%
ara	78.3	6.5k	1.8	78.6	–1%	100%	**79.0**	–31%	0%	**79.0**	–9%	100%
ita	91.1	5.2k	1.8	91.2	9%	59%	**91.5**	–9%	0%	**91.5**	10%	59%
por	90.3	8.3k	5.0	90.4	10%	46%	**90.5**	–22%	0%	**90.5**	10%	46%
heb	90.6	3.5k	3.1	90.5	23%	31%	**90.8**	11%	0%	90.6	–3%	31%
eng	92.7	5.4k	3.1	92.6	–21%	7%	**92.8**	–3%	0%	**92.8**	–11%	7%
cmn	**89.4**	10.4k	6.0	**89.4**	–16%	0%	**89.4**	–12%	0%	**89.4**	0%	0%

correlation coefficient of 0.748 for `agr`, demonstrating that improvement due solely to agreement modeling correlates strongly with the amount of morphological information in the data. The previous error reduction data were likely polluted by improvement due to capturing the PPL information; the relationship of this improvement to morphological complexity is unknown. Correlation for `orig` and `agr+orig` is still moderate. The correlation coefficients for experiments both with and without the PPL feature appear in Table 5.5.

5.5 Weights of Original and Agreement Features

Table 5.6 displays the number of morphology features from each feature set appearing in the highest-weighted 1,000 features (from a single cross-validation fold) when pars-

Table 5.5: Correlation coefficient (Pearson's r) between error reduction relative to `no-morph` and average number of morphological attributes per token, and p-value indicating level of statistical signficance.

Feature configuration	r **(no PPL feature)**	r **(with PPL feature)**
`orig`	0.608, $p < 0.01$	0.506, $p < 0.02$
`agr`	0.560, $p < 0.01$	0.748, $p < 0.01$
`agr+orig`	0.428, $p < 0.10$	0.621, $p < 0.01$

ing the 9,000-sentence Czech and English datasets, with the PPL feature included. In the Czech data, the number of highly-weighted agreement features seen with `agr` is far more than the number of highly-weighted original features with `orig`. Furthermore, when running `agr+orig`, there are again many more highly-weighted agreement than original features, even though the total number of agreement features (10,000) is several orders of magnitude less than the total number of original features (26 million). This indicates that the agreement features contribute far more to parse scores than do the original features.

When running on the English data, no `orig` features appear in the top 1,000 features with any feature configuration. Because there is so little morphological information in the English data (averaging 0.4 attributes per token, vs. 2.8 for the Czech data), the `orig` set contributes far less to the total feature counts – approximately 230,000 features – than it does for Czech.

5.6 Including All vs. Some Attributes

The Czech treebank includes morphological annotations for a variety of attributes: person, number, gender, case, tense, degree of comparison, negation, and voice. I generated a version of this dataset including only person, number, gender, and case features – a subset commonly involved in agreement relationships – and tested the parser on it in order to determine roughly which attributes were contributing to the

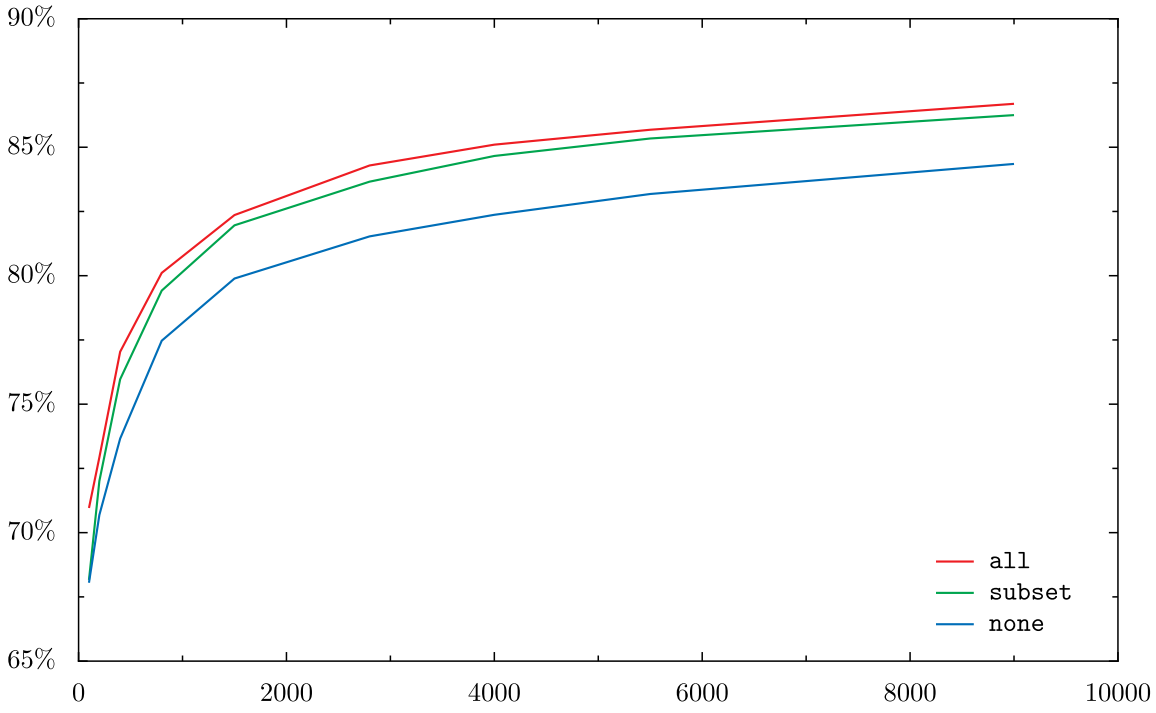
Table 5.6: Number of features from each set appearing in 1,000 highest-weighted features when parsing 9,000-sentence Czech and English datasets, with PPL feature included, and total number of features (including non-morphological features).

Language	Feature configuration	`orig` features	`agr` features	PPL features	Total features
Czech	`no-morph`	0	0	318	4,801,932
	`orig`	16	0	303	26,563,130
	`agr`	0	236	278	4,811,893
	`agr+orig`	2	178	277	26,573,091
English	`no-morph`	0	0	153	3,152,825
	`orig`	0	0	155	3,387,652
	`agr`	0	54	146	3,153,609
	`agr+orig`	0	57	148	3,388,436

improvement in performance.

Figure 5.7 shows the results of parsing at all dataset sizes with the `agr` feature configuration using three different groups of morphological attributes: none (`none`); only person, number, gender, and case (`subset`); and all attributes (`all`). The PPL feature was included in all three trials. On all but the smallest datasets (those of less than 1,000 sentences), the increase in accuracy from `none` to `subset` was about 80% of the increase from `none` to `all`. This indicates that for this treebank, the person, number, gender, and case attributes account for most of the performance increase when using the `agr` features, but the other attributes contribute as well. The three curves are roughly parallel, demonstrating that except on very small datasets, the contributions of the various subsets of attributes are independent of dataset size.

Table 5.7 itemizes the `agr` features appearing in the 1,000 highest-weighted features from a single cross-validation fold of the `all` test (using the 9,000-sentence dataset), revealing which attributes are the most useful to the parser. Five attributes had multiple labeled agreement features in this group, indicating that for each of those attributes, the parser discovered multiple commonly occurring relationships between agreeing parts of speech.



	100	200	400	800	1500	2800	4000	5500	9000
`none`	68.1	70.7	73.7	77.5	79.9	81.5	82.4	83.2	84.4
`subset`	68.2	72.0	76.0	79.4	82.0	83.7	84.7	85.3	86.3
`all`	71.0	72.9	77.0	80.1	82.4	84.3	85.1	85.7	86.7

Figure 5.7: Unlabeled accuracy when parsing Czech data including no morphological information, a subset of morphological information, and all morphological information. The `agr` feature configuration was used and the PPL feature was included.

The counts for asymmetric features are significantly higher than those for symmetric. This is at least partly because more of them are generated; since the asymmetric features include both attribute and value, there are many more possible features of this type. That is, given the head and dependent parts of speech, for an attribute marked on both, only two unlabeled symmetric features can be generated (agreement and disagreement). If an attribute is marked only on one or the other, however, two unlabeled asymmetric features can be generated for each possible value of the attribute, depending on whether it is marked on the head or the dependent.

Table 5.7: Number of each type of agreement features appearing in the 1,000 highest-weighted features in Czech when including all morphological attributes. PPL feature is included.

Attribute	Agreement features	Disagreement features	Asymmetric features	Total
person	0	1	13	14
number	10	0	27	37
gender	3	3	12	18
case	11	3	72	86
voice	2	1	16	19
tense	1	1	7	9
negation	10	1	20	31
degree of comparison	0	0	2	2
possessor number	0	0	0	0
possessor gender	0	0	0	0
semantic class	1	1	16	18
variant	0	0	2	2

5.7 Summary

The `agr` feature set increased unlabeled parsing accuracy on all treebanks which include any morphological information. Using both the `agr` and `orig` sets increased it slightly more in a few cases, but in most, `agr` alone performed better. Run times and feature counts for `agr` were significantly lower than those for `orig`. Improvement was roughly independent of dataset size for all but the smallest datasets. I found that the agreement feature set is less sensitive than the original to inaccurate annotations in automatically tagged data.

Inspecting the highly-weighted features from `agr` trials revealed that some of the most important features were labeled asymmetric features. Adding to the parser a feature template which includes head and dependent POS and dependency label, but no morphological information, indicated that a portion of the improvement seen with the `agr` features was due to their encapsulation of that information, not to their modeling of agreement. With this correction made, however, a stronger correlation

between error reduction and quantity of morphological information emerged.

Further examination of feature weights indicated that `agr` features were in general more useful for parsing than `orig` features. Finally, running the new feature set on Czech data including only attributes which commonly participated in agreement showed that both agreement and non-agreement (asymmetric) relationships contribute to parsing accuracy.

Chapter 6
FUTURE WORK AND CONCLUSION

6.1 Future Work

This research suggests several directions for future work on using morphological information in dependency parsing.

One possibility involves more careful normalization of treebanks. For instance, if an adjective can agree with either a masculine or a feminine noun, annotating the adjective with both `gen=M` and `gen=F` (rather than `gen=M/F`, or `gen=X`, as annotated in several of the treebanks used here) would ensure that agreement with a noun of either gender would be captured by the agreement features. Other experiments could include filtering out morphological information, perhaps predicated on part-of-speech, or feature selection based on which features are most informative.

Feature selection or pruning on the morphological features generated by the unaltered MSTParser system could be informative; it is possible that the modified system performs better primarily because it generates fewer features and does not overwhelm the learning algorithm with less informative features. The agreement features described here could easily be added to other systems, such as MaltParser. A final direction for future work is developing a metric to measure the degree of word order flexibility in a treebank, in order to explore the extent to which the degree of improvement achieved by the agreement model correlates with word order flexibility.

6.2 Conclusion

This work describes a simple, language-independent model of agreement designed to better leverage morphological information in dependency parsing. Testing on a variety

of treebanks containing different amounts of morphological information revealed that this modification achieved substantial improvements in parsing accuracy. This was due partly to the agreement features capturing non-morphological information, but even when this was compensated for, the new feature set improved accuracy on every treebank while reducing feature counts and run times significantly. Furthermore, the modifications were relatively simple, affecting only two files and around one hundred lines of code.

The agreement model was originally intended to compensate for lower parsing accuracy on morphologically rich languages, which tend to use morphology to encode the syntactic roles encoded by word order in more analytic languages. After correcting for the effect of the PPL feature, I found a significantly stronger correlation between accuracy improvement and quantity of morphological annotation in a treebank when using the new feature set, indicating that the new features more concisely capture and leverage the morphology of a language. Despite this, the agreement model improved performance on every treebank with any amount of morphological information.

The agreement model was tested on treebanks which differ widely in annotation guidelines. Because of this, variables such as the amount of morphological information included and the treatment of non-projective parses and coordination could affect parsing performance. As the model was intended as a first pass, I did not delve into these factors.

However, this is part of the strength of the approach. Significant performance gains were achieved without any detailed knowledge of either the morphology of the languages used or of the annotation guidelines of the treebanks used. I hope that these results will encourage similarly linguistically motivated design in future systems. This case study provides strong evidence that incorporating linguistic knowledge into NLP systems does not preclude language independence, and may actually enhance it, by leveling performance across languages differing in morphological complexity.

REFERENCES

Aduriz, I., Aranzabe, M., Arriola, J., Atutxa, A., de Ilarraza, A., Garmendia, A., et al. (2003). Construction of a Basque dependency treebank. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT 2003)* (pp. 201–204).

Afonso, S., Bick, E., Haber, R., & Santos, D. (2002). Floresta Sintá(c)tica: A treebank for Portuguese. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002)* (p. 1698).

Ambati, B., Husain, S., Nivre, J., & Sangal, R. (2010). On the role of morphosyntactic features in Hindi dependency parsing. In *Proceedings of the First Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)* (pp. 94–102).

Attardi, G., Dell'Orletta, F., Simi, M., Chanev, A., & Ciaramita, M. (2007). Multilingual dependency parsing and domain adaptation using DeSR. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)* (pp. 1112–1118).

Bamman, D., & Crane, G. (2006). The design and use of a Latin dependency treebank. In *Proceedings of the Fifth International Workshop on Treebanks and Linguistic Theories (TLT 2006)* (pp. 67–78).

Bamman, D., Mambrini, F., & Crane, G. (2009). An ownership model of annotation: The Ancient Greek Dependency Treebank. In *Proceedings of the Eighth International Workshop on Treebanks and Linguistic Theories (TLT8)* (pp. 5–15).

Bender, E. (2011). On achieving and evaluating language-independence in NLP.

Linguistic Issues in Language Technology: Special Issue on Interaction of Linguistics and Computational Linguistics, *6*(3), 1–26.

Bengoetxea, K., & Gojenola, K. (2010). Application of different techniques to dependency parsing of Basque. In *Proceedings of the First Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)* (pp. 31–39).

Bhatt, R., Narasimhan, B., Palmer, M., Rambow, O., Sharma, D., & Xia, F. (2009). A multi-representational and multi-layered treebank for Hindi/Urdu. In *Proceedings of the Third Linguistic Annotation Workshop (LAW III)* (pp. 186–189).

Bick, E. (2006). LingPars, a linguistically inspired, language-independent machine learner for dependency treebanks. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)* (pp. 171–175).

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

Bosco, C., Lombardo, V., Vassallo, D., & Lesmo, L. (2000). Building a treebank for Italian: a data-driven annotation schema. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC 2000)* (pp. 99–106).

Brants, T., Skut, W., & Uszkoreit, H. (1999). Syntactic annotation of a German newspaper corpus. *Treebanks: Building and using parsed corpora*, *20*, 73.

Buchholz, S., & Marsi, E. (2006). CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)* (pp. 149–164).

Carreras, X., Surdeanu, M., & Marquez, L. (2006). Projective dependency parsing with perceptron. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)* (pp. 181–185).

Chang, M., Do, Q., & Roth, D. (2006). A pipeline model for bottom-up dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)* (pp. 186–190).

Chen, W., Zhang, Y., & Isahara, H. (2007). A two-stage parser for multilingual dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)* (pp. 1129–1133).

Chu, Y., & Liu, T. (1965). On the shortest arborescence of a directed graph. *Science Sinica*, *14*(1396-1400), 270.

Collins, M., Ramshaw, L., Hajič, J., & Tillmann, C. (1999). A statistical parser for Czech. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 505–512).

Corbett, G. (2006). *Agreement*. Cambridge University Press.

Corston-Oliver, S., & Aue, A. (2006). Dependency parsing with reference to Slovene, Spanish and Swedish. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)* (pp. 196–200).

Covington, M. (2001). A fundamental algorithm for dependency parsing. In *Proceedings of the 39th Annual Association for Computing Machinery (ACM) Southeast Conference* (pp. 95–102).

Cowan, B., & Collins, M. (2005). Morphology and reranking for the statistical parsing of Spanish. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)* (pp. 795–802).

Dreyer, M., Smith, D., & Smith, N. (2006). Vine parsing and minimum risk reranking for speed and precision. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)* (pp. 201–205).

Duan, X., Zhao, J., & Xu, B. (2007). Probabilistic parsing action models for multilingual dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)* (pp. 940–946).

Džeroski, S., Erjavec, T., Ledinek, N., Pajas, P., Žabokrtsky, Z., & Žele, A. (2006).

Towards a Slovene dependency treebank. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006).*

Earley, J. (1970). An efficient context-free parsing algorithm. *Communications of the Association for Computing Machinery (ACM)*, *13*(2), 94–102.

Edmonds, J. (1968). *Optimum branchings.* National Bureau of Standards.

Eisner, J. (1996). Three new probabilistic models for dependency parsing: An exploration. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)* (Vol. 1, pp. 340–345).

Eryiğit, G., Nivre, J., & Oflazer, K. (2008). Dependency parsing of Turkish. *Computational Linguistics*, *34*(3), 357–389.

Goldberg, Y. (2011). *Automatic Syntactic Processing of Modern Hebrew.* Unpublished doctoral dissertation, Ben Gurion University.

Goldberg, Y., & Elhadad, M. (2009). Hebrew dependency parsing: Initial results. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)* (pp. 129–133).

Goldberg, Y., & Elhadad, M. (2010). Easy-first dependency parsing of Modern Hebrew. In *Proceedings of the First Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)* (pp. 103–107).

Hajic, J., Smrz, O., Zemánek, P., Šnaidauf, J., & Beška, E. (2004). Prague Arabic dependency treebank: Development in data and tools. In *Proceedings of the NEMLAR International Conference on Arabic Language Resources and Tools* (pp. 110–117).

Hajič, J. (1998). Building a syntactically annotated corpus: The Prague Dependency Treebank. In E. Hajičová (Ed.), *Issues of Valency and Meaning: Studies in Honor of Jarmila Panevová* (pp. 12–19). Prague Karolinum, Charles University Press.

Hall, J., Nilsson, J., Nivre, J., Eryiğit, G., Megyesi, B., Nilsson, M., et al. (2007). Single Malt or blended? A study in multilingual parser optimization. In *Proceedings*

of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007) (pp. 933–939).

Haverinen, K., Viljanen, T., Laippala, V., Kohonen, S., Ginter, F., & Salakoski, T. (2010). Treebanking Finnish. In *Proceedings of the Ninth International Workshop on Treebanks and Linguistic Theories (TLT9* (Vol. 9, pp. 79–90).

Johansson, R., & Nugues, P. (2006). Investigating multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)* (pp. 206–210).

Karlsson, F. (1990). Constraint grammar as a framework for parsing running text. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING 1990)* (Vol. 3, pp. 168–173).

Kasami, T. (1965). *An efficient recognition and syntax analysis algorithm for context-free languages* (Tech. Rep.). DTIC Document.

Kromann, M. (2003). The Danish Dependency Treebank and the DTAG treebank tool. In *Proceedings of the Second Workshop on Treebanks and Linguistic Theories (TLT 2003)* (pp. 217–220).

Lee, Y., Haghighi, A., & Barzilay, R. (2011). Modeling syntactic context improves morphological segmentation. In *Proceedings of the 15th Conference on Computational Natural Language Learning (CONLL-2011)* (pp. 1–9).

Liu, T., Ma, J., Zhu, H., & Li, S. (2006). Dependency parsing based on dynamic local optimization. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)* (pp. 211–215).

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

Martı, M., Taulé, M., Márquez, L., & Bertran, M. (2007). *CESS-ECE: A multilingual and multilevel annotated corpus.* Available from `http://www.lsi.upc.edu/`

~mbertran/cess-ece/

Marton, Y., Habash, N., & Rambow, O. (2010). Improving Arabic dependency parsing with lexical and inflectional morphological features. In *Proceedings of the First Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)* (pp. 13–21).

Marton, Y., Habash, N., & Rambow, O. (2011). Improving Arabic dependency parsing with form-based and functional morphological features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL HLT 2011)* (pp. 1586–1596).

Maruyama, H. (1990). Structural disambiguation with constraint propagation. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 31–38).

McDonald, R., Crammer, K., & Pereira, F. (2005). Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 91–98).

McDonald, R., Lerman, K., & Pereira, F. (2006). Multilingual dependency analysis with a two-stage discriminative parser. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)* (pp. 216–220).

McDonald, R., Pereira, F., Ribarov, K., & Hajič, J. (2005). Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)* (pp. 523–530).

Minkov, E., Toutanova, K., & Suzuki, H. (2007). Generating complex morphology for machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)* (Vol. 45, p. 128).

Nakagawa, T. (2007). Multilingual dependency parsing using global features. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-*

CoNLL 2007) (pp. 952–956).

Nguyen, L., Shimazu, A., Nguyen, P., & Phan, X. (2007). A multilingual dependency analysis system using online passive-aggressive learning. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)* (pp. 1149–1155).

Nivre, J. (2009). Parsing Indian languages with MaltParser. In *Proceedings of the Seventh International Conference on Natural Language Processing (ICON 2009) NLP Tools Contest* (pp. 12–18).

Nivre, J., Boguslavsky, I., & Iomdin, L. (2008). Parsing the SynTagRus treebank of Russian. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)* (Vol. 1, pp. 641–648).

Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., et al. (2007). CoNLL 2007 shared task on dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007).*

Nivre, J., Hall, J., & Nilsson, J. (2006). Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)* (Vol. 6, pp. 2216–2219).

Nivre, J., Hall, J., Nilsson, J., Eryiğit, G., & Marinov, S. (2006). Labeled pseudo-projective dependency parsing with support vector machines. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)* (pp. 221–225).

Oflazer, K., Say, B., Hakkani-Tür, D., & Tür, G. (2003). Building a Turkish treebank. *Text, Speech, and Language Technology*, 261–277.

Øvrelid, L., & Nivre, J. (2007). When word order and part-of-speech tags are not enough–Swedish dependency parsing with rich linguistic features. In *Proceedings of the International Conference on Recent Advances in Natural Language*

Processing (RANLP) (pp. 447–451).

Petrov, S., Das, D., & McDonald, R. (2011). A universal part-of-speech tagset. *Arxiv preprint ArXiv:1104.2086*.

Rajkumar, R., & White, M. (2010). Designing agreement features for realization ranking. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)* (pp. 1032–1040).

Ramasamy, L., & Žabokrtský, Z. (2011). Tamil dependency parsing: Results using rule based and corpus based approaches. In *Proceedings of the 12th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2011)* (Vol. 1, pp. 82–95). Berlin, Heidelberg: Springer-Verlag. Available from `http://portal.acm.org/citation.cfm?id=1964799.1964808`

Riedel, S., & Clarke, J. (2006). Incremental integer linear programming for non-projective dependency parsing. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)* (pp. 129–137).

Samuelsson, C. (2000). A statistical theory of dependency syntax. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)* (pp. 684–690).

Schiehlen, M., & Spranger, K. (2007). Global learning of labelled dependency trees. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)* (pp. 1156–1160).

Siewierska, A. (1998). Variation in major constituent order: A global and a European perspective. In A. Siewierska (Ed.), *Constituent order in the Languages of Europe* (pp. 475–551). Mouton De Gruyter.

Simov, K., Osenova, P., Simov, A., & Kouylekov, M. (2004). Design and implementation of the Bulgarian HPSG-based treebank. *Research on Language & Computation*, *2*(4), 495–522.

Tarjan, R. (1977). Finding optimum branchings. *Networks*, *7*(1), 25–35.

Tesnière, L. (1959). *Éléments de syntaxe structurale* [Elements of structural syntax]. Klincksieck.

Titov, I., & Henderson, J. (2007). Fast and robust multilingual dependency parsing with a generative latent variable model. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)* (pp. 947–951).

Tsarfaty, R., & Sima'an, K. (2010). Modeling morphosyntactic agreement in constituency-based parsing of Modern Hebrew. In *Proceedings of the First Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)* (pp. 40–48).

Van der Beek, L., Bouma, G., Malouf, R., & Van Noord, G. (2002). The Alpino dependency treebank. *Language and Computers*, *45*(1), 8–22.

Vincze, V., Szauter, D., Almási, A., Móra, G., Alexin, Z., & Csirik, J. (2010). Hungarian dependency treebank. In *Proceedings of the Seventh Conference on Language Resources and Evaluation (LREC 2010).*

Wu, Y., Yang, J., & Lee, Y. (2007). Multilingual deterministic dependency parsing framework using modified finite Newton method support vector machines. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)* (pp. 1175–1181).

Xue, N., Xia, F., Chiou, F., & Palmer, M. (2005). The Penn Chinese Treebank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, *11*(2), 207–238.

Younger, D. (1967). Recognition and parsing of context-free languages in time n^3. *Information and control*, *10*(2), 189–208.

Yuret, D. (2006). Dependency parsing as a classification problem. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)* (pp. 246–250).

Appendix A

IMPLEMENTATION OF AGREEMENT FEATURES

Implementing the agreement feature set required making alterations to `mstparser/DependencyPipe.java`. The original morphological features are added in the code in Listing A.1; this was replaced with the code in Listing A.2 to generate the unlabeled agreement features. Labeled agreement features were added by inserting the code in Listing A.3 at the end of the `addLabeledFeatures` method.

```
for (int i=0; i<instance.feats[headIndex].length; i++) {
   for (int j=0; j<instance.feats[childIndex].length; j++) {
      addTwoObsFeatures("FF"+i+"*"+j,
         instance.forms[headIndex],
         instance.feats[headIndex][i],
         instance.forms[childIndex],
         instance.feats[childIndex][j],
         attDist, fv);

      addTwoObsFeatures("LF"+i+"*"+j,
         instance.lemmas[headIndex],
         instance.feats[headIndex][i],
         instance.lemmas[childIndex],
         instance.feats[childIndex][j],
         attDist, fv);
   }
}
```

Listing A.1: Source code for original morphological features.

```
boolean headAttsMatched[] = new boolean[instance.feats[headIndex].length];
boolean depAttsMatched[] = new boolean[instance.feats[childIndex].length];

String hPOS = posA[headIndex];                          // grab head CPOS
String dPOS = posA[childIndex];                         // grab dep CPOS

for (int i=0; i<instance.feats[headIndex].length; i++) {       // for each head attr
   for (int j=0; j<instance.feats[childIndex].length; j++) {   // for each dep attr
      String headItem = instance.feats[headIndex][i];          // "item": attr=val
      String depItem = instance.feats[childIndex][j];

      if (headItem.contains("=") && depItem.contains("=")) {      // if not "_"
         String headAtt = instance.feats[headIndex][i].split("=")[0];
         String depAtt = instance.feats[childIndex][j].split("=")[0];
         String headVal = instance.feats[headIndex][i].split("=")[1];
         String depVal = instance.feats[childIndex][j].split("=")[1];

         if (depAtt.equals(headAtt)) {              // if same attr
            headAttsMatched[i] = true;              // found a match for this attr
            depAttsMatched[j] = true;

            if (depVal.equals(headVal))          // if same value, add "agrees"
               add(headAtt+"_agrees,head="+hPOS+",dep="+dPOS,fv);
            else                                 // if different, add "disagrees"
               add(headAtt+"_disagrees,head="+hPOS+",dep="+dPOS,fv);
         }
      }
   }
}

for (int i=0; i<headAttsMatched.length; i++)                // for each head attr
   if (!headAttsMatched[i]) {                               // if unmatched
      String headItem = instance.feats[headIndex][i];       // add asymmetric
      add("head_"+headItem+",head="+hPOS+",dep="+dPOS,fv);
   }

for (int i=0; i<depAttsMatched.length; i++)                 // for each dep attr
   if (!depAttsMatched[i]) {                                // if unmatched
      String depItem = instance.feats[childIndex][i];       // add asymmetric
      add("dep_"+depItem+",head="+hPOS+",dep="+dPOS,fv);
   }
```

Listing A.2: Source code for unlabeled agreement features.

```
if (childFeatures && (instance.heads[word] != -1)) {
   String[] headFeats = instance.feats[instance.heads[word]];
   String[] childFeats = instance.feats[word];

   String hPOS = instance.cpostags[instance.heads[word]];      // grab head CPOS
   String dPOS = instance.cpostags[word];                      // grab dep CPOS

   boolean headAttsMatched[] = new boolean[headFeats.length];
   boolean depAttsMatched[] = new boolean[childFeats.length];

   for (int i = 0; i < childFeats.length; i++) {               // for each head attr
      for (int j = 0; j < headFeats.length; j++) {             // for each dep attr
         if (headFeats[j].contains("=") && childFeats[i].contains("=")) {
            String headAtt = headFeats[j].split("=")[0];
            String depAtt = childFeats[i].split("=")[0];
            String headVal = headFeats[j].split("=")[1];
            String depVal = childFeats[i].split("=")[1];

            if (depAtt.equals(headAtt)) {                      // if same attribute
               headAttsMatched[j] = true;                      // found a match for this attr
               depAttsMatched[i] = true;

               if (depVal.equals(headVal))                     // if same value, add "agrees"
                  add(depAtt+"_agrees&label="+type+",head="+hPOS+",dep="+dPOS, fv);
               else                                            // if different, add "disagrees"
                  add(depAtt+"_disagrees&label="+type+",head="+hPOS+",dep="+dPOS, fv);
            }
         }
      }
   }

   for (int i=0; i<headAttsMatched.length; i++)                // for each head attr
      if (!headAttsMatched[i]) {                               // if unmatched
         String headItem = headFeats[i];                       // add asymmetric
         add("head_"+headItem+",head="+hPOS+",dep="+dPOS+",label="+type, fv);
      }

   for (int i=0; i<depAttsMatched.length; i++)                 // for each dep attr
      if (!depAttsMatched[i]) {                                // if unmatched
         String depItem = childFeats[i];                       // add asymmetric
         add("dep_"+depItem+",head="+hPOS+",dep="+dPOS+",label="+type, fv);
      }
}
```

Listing A.3: Source code for labeled agreement features.

Appendix B

ORIGINAL TO UNIVERSAL POS TAG MAPPINGS

The universal tagset suggested by Petrov et al. (2011) includes twelve tags: `NOUN` (noun), `PRON` (pronoun), `VERB` (verb), `ADJ` (adjective), `ADV` (adverb), `ADP` (adposition), `CONJ` (conjunction), `PRT` (particle), `NUM` (numeral), `.` (punctuation), and `X` (unknown). The authors also included mappings to generate universal tags from the tags in the Arabic, Basque, Bulgarian, Catalan, Chinese, Czech, Danish, Dutch, English, German, Hungarian, Portuguese, Slovene, and Turkish treebanks.

Mappings used for the Finnish, Ancient Greek and Latin, Hebrew, Italian, and Tamil treebanks were derived from the treebank documentation and are included here.

Table B.1: Mapping for Finnish treebank (Turku).

Original tag	Normalized tag
A	ADJ
ABBR	X
AD-A	ADJ
ADV	ADV
ART	X
C	CONJ
DV-MA	VERB
FORGN	X
INTJ	X
N	NOUN
NON-TWOL	X
NUM	NUM
PCP1	VERB
PCP2	VERB
PP	ADP
PRON	PRON
PROP	NOUN
PSP	ADP
PUNCT	.
Q	DET
TrunCo	X
V	VERB

Table B.2: Mapping for Ancient Greek (AGDT) and Latin (LDT) treebanks.

Original tag	Normalized tag
n	NOUN
v	VERB
t	VERB
a	ADJ
d	ADV
l	DET
g	PRT
c	CONJ
r	ADP
p	PRON
m	NUM
i	X
e	X
u	.

Table B.3: Mapping for Modern Hebrew treebank (DepTB).

Original tag	Normalized tag
!!MISS!!	X
!!SOME_!!	X
!!UNK!!	X
!!ZVL!!	X
ADVERB	PRT
AUX	VERB
AT	PRT
CC	CONJ
CC-COORD	CONJ
CD	NUM
CDT	NUM
CONJ	CONJ
COP	VERB
DEF	PRT
DT	DET
DTT	DET
H	PRT
IN	ADP
IN$2	ADP
INTJ	X
JJ	ADJ
JJT	ADJ
MD	VERB
NN	NOUN
NNP	NOUN
NNT	NOUN
P	PRT
POS	PRT
PREPOSITION	ADP
PRP	PRON
PUNC	.
QW	ADV
RB	ADV
RBR	ADV
REL	CONJ
VB	VERB
VB-*	VERB
WDT	DET

Table B.4: Mapping for Hindi treebank (HyDT).

Original tag	Normalized tag
ADJ	ADJ
ADV	ADV
AVY	PRT
CC	CONJ
DEM	DET
INJ	X
INTF	ADV
JJ	ADJ
JJC	ADJ
N	NOUN
NEG	ADV
NN	NOUN
NNC	NOUN
NNP	NOUN
NNPC	NOUN
NST	NOUN
NSTC	NOUN
NUM	NUM
PN	PRON
PRP	PRON
PRPC	PRON
PSP	ADP
PUNC	.
QC	NUM
QCC	NUM
QF	DET
QFC	DET
QO	ADJ
RB	ADV
RBC	ADV
RP	PRT
SYM	.
UNK	X
UT	X
V	VERB
VAUX	VERB
VM	VERB
VMC	VERB
WQ	ADV
XC	X

Table B.5: Mapping for Italian treebank (TDT).

Original tag	Normalized tag
ADJ	ADJ
ADV	ADV
ART	DET
CONJ	CONJ
DATE	NUM
HOUR	NUM
INTERJ	X
NOUN	NOUN
NUM	NUM
PHRAS	X
PREDET	DET
PREP	ADP
PRON	PRON
PUNCT	.
SPECIAL	X
VERB	VERB

Table B.6: Mapping for Tamil treebank (TamilTB).

Original tag	Normalized tag
A	ADV
C	CONJ
D	DET
I	X
J	ADJ
N	NOUN
P	ADP
Q	ADJ
R	PRON
T	PART
U	NUM
V	VERB
X	X
Z	.

Appendix C

ENGLISH POS TAG TO MORPHOLOGICAL ANNOTATION MAPPING

The following mapping was used to generate morphological attributes for the English data, using the original Penn Treebank POS tags. Pronoun attributes are dependent on word form as well as POS tag.

Table C.1: POS tag to morphological annotation mapping for English treebank (Penn).

POS tag	Word form	Morphological attributes generated
NN, NNP	(any)	pernum=3sg
NNS, NNPS	(any)	pernum=3pl
PRP	I	pernum=1sg, case=nom
	me, myself	pernum=1sg, case=acc
	you	pernum=2
	he, she	pernum=3sg, case=nom
	him, himself, her, herself	pernum=3sg, case=acc
	it	pernum=1sg
	itself	pernum=3sg, case=acc
	we	pernum=1pl, case=nom
	us, ourselves	pernum=1pl, case=acc
	they	pernum=3pl, case=nom
	themselves	pernum=3pl, case=acc
PRP$	my	pernum=1sg, case=gen
	your	pernum=2, case=gen
	his, her, its	pernum=3sg, case=gen
	our	pernum=1pl, case=gen
	their	pernum=3pl, case=gen
VBP	(any)	pernum=non3sg
VBZ	(any)	pernum=3sg

Appendix D

MORPHOLOGICAL INFORMATION IN TREEBANKS

Each treebank was annotated with different morphological information, based on the properties of the language. Many included annotations of the form `attribute=value`; for those which included only values, I referred to the treebank documentation to associate values with attributes. The following table summarizes the information in each treebank.

Table D.1: Attributes annotated in each treebank, with all possible values for each.

Language	Attribute	Values
Arabic	case	1 2 4 F
	def	C D F I P R
	gen	U S M F
	mood	I S E D T
	num	P S U D N
	pers	1 2 3 O N
	voice	P
Basque	advtype	GRAD
	allo	NO TO
	asp	BURU EZBU GERO PNT
	case	ABL ABS ABU ABZ ALA BNK DAT DES DESK ERG GEL GEN INE INS MOT PAR PRO SOZ
	def	DEF INDEF
	deg	GEHI IND KONP SUP
	mod	EGI ZIU
	mod-tense	A1 A3 A4 A5 B1 B2 B3 B4 B5 B6 B7 B8
	nor	GU HAIEK HI HK HU HURA NI ZU ZUEK
	nori	GURI HAIEI HARI HI HIRI-NO HIRI-TO HK HU NI NIRI ZU ZURI
	nork	GU GUK HAIEK-K HARK HI HIK HIK-NO HIK-TO HK HU NI NIK ZU ZUEK-K ZUK
	num	P PH S
	per	GU HAIEK HI NI ZU ZUEK
	rel	AURK BALD DENB EMEN ERLT ESPL HAUT HELB KAUS KONPL KONT MOD MOD/DENB MOS ONDO ZHG
	subcat	ADK ARR BAN FAK MEN ORD
	vtype	ADIZE ADOIN PART

Language	Attribute	Values
Bulgarian	aspect	P
	case	A D DP N V
	def	D F H I
	form	EXT F S
	gen	F M N
	imtrans	I T
	mood	I U Z
	num	P PIA_TANTUM S T
	pers	1 2 3
	ref	A E L M MP OP P Q R T
	tense	M O R
	trans	I T
	type	AUX
	vform	C G
	voice	A V
Catalan	case	A D N O
	for	C S
	gen	C F M N
	mod	G I M N P S
	num	N O P S
	pari	P
	per	1 2 3
	pos	P S
	tmp	C F I P S
Czech	Cas	1 2 3 4 5 6 7 X
	Gen	F H I M N Q T X Y Z
	Gra	1 2 3
	Neg	A N
	Num	D P S W X
	PGe	F M X Z
	PNu	P S X
	Per	1 2 3 X
	Sem	E G K R S Y M
	Ten	F P R X
	Var	1 2 3 4 5 6 7 8 9
	Voi	A P
Danish	case	GEN NOM UNMARKED
	def	DEF DEF/INDEF INDEF
	definiteness	DEF DEF/INDEF INDEF
	degree	ABS COMP POS SUP UNMARKED
	gender	COMMON COMMON/NEUTER NEUTER
	mood	GERUND IMPER INDIC INFIN PARTIC
	number	PLUR SING SING/PLUR
	person	1 2 3
	possessor	PLUR SING SING/PLUR
	reflexive	NO YES YES/NO
	register	FORMAL OBSOLETE POLITE UNMARKED
	tense	PAST PRESENT
	transcat	ADJECT ADJECT/ADVERB/UNMARKED ADVERB ADVERBIAL UNMARKED
	voice	ACTIVE PASSIVE

Language	Attribute	Values
Dutch	case	DAT DAT-OR-ACC GEN NOM NONE
	clause	FIN INF
	deg	COMP POS SUP
	form	GEN INFL PL UNINFL
	func	DEM ER INDET INTER NONE REL
	num	DET INDET PL SG SG-OR-PL
	numgen	NEUT NEUT-OR-NONNEUT NONNEUT NONNEUT-OR-PL
	per	1 1OR2OR3 2 3
	tense	CONJ IMP INF PAST-IMPERF PAST-PART PRES-IMPERF PRES-PART
	type	AUX AUX-OR-COP CARD COM COMB COORD DEF DEMON INDEF INDET INTRANS ORD PER POSS POST PRE PROP RECIP REFL REL SUBORD TRANS WH
	use	ADV ATTR DEELADV DEELV IND NORM PRON
Finnish	Case	ABE ABL ADE ALL CMT ELA ESS GEN ILL INE INS NOM PTV TRA
	Clitic	HAN KA KAAN KO KIN PA S
	Comp	CMP POS SUP
	Inf	INF1 INF2 INF3
	Mood	COND IMPV POTN
	Neg	NEG NEGV
	Num	PL SG
	Person	PE4 PL1 PL2 PL3 SG1 SG2 SG3
	Poss	1PL 1SG 2PL 2SG 3
	Tense	PAST PRES
	Voice	ACT PSS
German	case	ACC DAT GEN NOM
	definite	NO YES
	degree	COMPARATIVE POSITIVE SUPERLATIVE
	flexion	MIXED STRONG WEAK
	gender	FEM MASC NEUT
	mood	INDICATIVE SUBJUNCTIVE
	number	PL SG
	person	FIRST SECOND THIRD
	tense	PAST PRESENT
Greek (ancient)	case	A D G N V
	deg	C S
	gen	F M N
	mood	I M N O P S
	num	D P S
	per	1 2 3
	tense	A F I L P R T
	voice	A E M P
Hebrew (modern)	gen	F M
	num	P S
	per	1 2 3
Hindi	cas	0 3 ANY D O
	gen	3 ADJ ANY F M N NUM PL PUNC SG
	num	3 AG ANY M PL SD SG SG3
	per	1 1H 2 2H 3 3H ANY D M O PL SG
	suf	0 00 WA ANY EM EMS GA HE HAI KA KA KAR KARA KE KO ME MEM NA NA_VALA NE O S SE WA YA YA1

Language	Attribute	Values
Hungarian	Cas	1 2 3 6 9 A B C D E F G H I L M N NONE P Q S T U W X Y
	Coord	P W
	Def	2 N Y
	Deg	C NONE P S
	Form	C D L R S
	Mood	C I M N
	Num	NONE P S
	NumP	NONE P S
	NumPd	NONE P S
	Per	1 2 3 NONE
	PerP	1 2 3 NONE
	Tense	P S
	Type	D F M O P Q R S T W
Italian	gen	F M
	num	PL SING
	per	1 2 3
Latin	case	A B D G L N V
	deg	C S
	gen	F M N
	mood	D G I M N P S U
	num	P S
	per	1 2 3
	tense	F I L P R T
	voice	A D P
Portuguese	case	ACC DAT NOM PIV
	gen	F M
	num	P S
	per	1
Slovene	Animate	NO YES
	Case	ACCUSATIVE DATIVE GENITIVE INSTRUMENTAL LOCATIVE NOMINATIVE
	Clitic	NO YES
	Definiteness	NO YES
	Degree	COMPARATIVE POSITIVE SUPERLATIVE
	Form	DIGIT LETTER
	Formation	COMPOUND SIMPLE
	Gender	FEMININE MASCULINE NEUTER
	Negative	NO YES
	Number	DUAL PLURAL SINGULAR
	Owner-Gender	FEMININE MASCULINE NEUTER
	Owner-Number	DUAL PLURAL SINGULAR
	Person	FIRST SECOND THIRD
	Referent-Type	PERSONAL POSSESSIVE
	Syntactic-Type	ADJECTIVAL NOMINAL
	Tense	FUTURE PAST PRESENT
	VForm	CONDITIONAL IMPERATIVE INDICATIVE INFINITIVE PARTICIPLE SUPINE
	Voice	ACTIVE PASSIVE

Language	Attribute	Values
Tamil	Cas	A D G I L N S
	Gen	A H M N
	Neg	A N
	Num	P S
	Per	1 2 3
	Ten	D F P T
	Voi	A P
Turkish	case	ABL ACC DAT GEN INS LOC NOM
	comp_mod	ABLE HASTILY STAY
	pccase	PCABL PCACC PCDAT PCGEN PCINS PCNOM
	pernum	A1PL A1SG A2PL A2SG A3PL A3SG
	pol	NEG POS
	poss	P1PL P1SG P2PL P2SG P3PL P3SG PNON
	tam	AOR COND DESR FUT IMP NARR NECES OPT PAST PRES PROG1 PROG2
	voice	CAUS PASS RECIP REFLEX

Appendix E

UNDERSCORE/ROOT FEATURE MODIFICATION TO MSTPARSER

In the original version of the parser, the underscores used to indicate the absence of morphological information are themselves treated as morphological attributes. Furthermore, multiple copies of a placeholder morphological attribute were added to the "root" token of each sentence; this seems to have been left over from a previous version of the software which required that each token have the same number of attributes.

Since the `agr` and `orig` feature sets handle the underscores differently, their inclusion meant that those sets performed differently on treebanks containing no morphological information. In order to rectify this, I made two small changes to the file `mstparser/io/CONLLReader.java`, as described in Listings E.1 and E.2.

Original:

```
feats[i+1] = info[5].split("\\|");
```

Revision:

```
if (!info[5].equals("_"))                  // if not "_"
   feats[i+1] = info[5].split("\\|");      // split into list
else                                       // otherwise
   feats[i+1] = new String[0];             // make empty list
```

Listing E.1: Source code for underscore modification.

These alterations affected performance negatively on some treebanks, and positively on others, with the absolute difference ranging from -0.48% to 0.19%. This is due to two competing consequences of the inclusion of the underscore and root features: the sheer quantity of features generated tended to overwhelm the learner and decrease accuracy,

Original:

```
feats[0] = new String[feats[1].length];
for (int i = 0; i< feats[1].length; i++)
   feats[0][i] = "<root-feat>"+i;
```

Revision:

```
feats[0] = new String[0];              // always add empty list
```

Listing E.2: Source code for root feature modification.

while the small amount of information gleaned from the association of certain tokens with the root tended to increase it. The effect is also less noticeable on treebanks with more morphological information, because those treebanks include fewer underscores. Accuracies on all treebanks with and without this modification are listed in Table E.1.

Table E.1: Unlabeled accuracy using the `orig` feature set on all treebanks at reference size, with and without underscore/root feature modification.

Language	Accuracy before modification	Accuracy after modification	Change in accuracy
hin	92.14	92.01	−0.13
hun	88.85	88.66	−0.19
ces	81.85	81.63	−0.22
tam	79.85	79.68	−0.17
slv	80.21	80.40	0.19
dan	88.66	88.37	−0.29
lat	65.45	64.97	−0.48
nld	89.01	89.04	0.03
eus	80.60	80.23	−0.37
bul	90.26	90.15	−0.11
grc	77.37	76.95	−0.42
deu	90.91	90.82	−0.09
fin	76.56	76.30	−0.26
tur	81.55	81.51	−0.04
cat	81.91	81.94	0.03
ara	78.29	78.06	−0.23
ita	89.13	88.91	−0.22
por	88.06	88.15	0.09
heb	84.24	84.15	−0.09
eng	88.08	88.01	−0.07
cmn	82.18	82.35	0.17

Appendix F

RAW DATA

Table F.1: Reference size, unlabeled accuracy, run time in seconds, and number of features in millions, for all treebanks containing morphological information. The highest accuracy for each treebank appears in bold face. Languages appear in decreasing order of average number of morphological attributes per token.

	no-morph			orig			agr			agr+orig		
Lang	UAC	time	feats	UAC	time	feats	UAC	time	feats	UAC	time	feats
hin	90.0	1.4k	1.6	92.0	3.1k	15.5	**93.8**	2.1k	1.6	93.0	3.5k	15.5
hun	87.9	4.6k	5.3	88.7	13.7k	41.6	**90.3**	5.0k	5.3	89.9	11.8k	41.6
ces	80.9	3.3k	4.8	81.6	5.6k	26.6	**85.5**	4.2k	4.8	84.5	7.0k	26.7
tam	79.0	0.1k	0.5	79.7	0.5k	2.0	**82.1**	0.2k	0.5	81.1	0.5k	2.0
slv	80.8	0.8k	1.0	80.4	1.7k	4.7	**81.8**	1.0k	1.0	80.8	1.9k	4.7
dan	87.8	2.0k	1.6	88.4	3.3k	5.8	**89.3**	2.4k	1.6	**89.3**	3.6k	5.8
lat	61.7	1.8k	1.6	65.0	2.7k	6.7	**70.3**	3.4k	1.7	68.6	3.8k	6.7
nld	88.2	2.0k	3.6	89.0	3.6k	13.4	**90.5**	2.3k	3.6	90.3	3.9k	13.4
eus	78.7	0.7k	1.7	80.2	1.2k	5.5	**82.3**	0.8k	1.7	**82.3**	1.2k	5.5
bul	89.9	1.7k	2.6	90.2	2.7k	8.4	**93.0**	2.0k	2.6	92.5	2.6k	8.4
grc	74.9	8.6k	3.8	77.0	11.6k	15.6	**80.7**	12.4k	3.8	79.5	14.5k	15.6
deu	90.0	0.9k	1.3	90.8	1.2k	3.6	**92.0**	0.9k	1.3	91.7	1.3k	3.6
fin	73.3	0.7k	2.4	76.3	1.3k	8.4	**79.1**	0.9k	2.4	78.7	1.4k	8.4
tur	80.2	1.2k	2.1	81.5	1.3k	5.8	81.6	1.1k	2.1	**81.7**	1.5k	5.8
cat	81.8	3.0k	2.5	81.9	3.1k	6.2	**84.9**	2.8k	2.5	84.0	3.0k	6.2
ara	77.6	5.4k	1.8	77.7	6.5k	3.7	**78.2**	4.9k	1.8	78.0	5.6k	3.7
ita	88.4	4.2k	1.8	88.9	4.1k	2.9	90.2	4.6k	1.8	**90.3**	4.5k	2.9
por	88.1	6.4k	5.0	88.2	7.5k	7.3	**89.0**	6.2k	5.0	88.9	8.1k	7.3
heb	87.4	4.3k	3.1	87.4	3.5k	4.1	**89.2**	3.6k	3.1	89.1	4.1k	4.1
eng	88.1	5.2k	3.1	88.0	5.4k	3.4	**90.6**	5.3k	3.1	**90.6**	4.7k	3.4
cmn	**82.4**	7.5k	6.0	**82.4**	10.3k	6.0	**82.4**	8.8k	6.0	**82.4**	9.3k	6.0

Table F.2: Reference size, unlabeled accuracy, run time in seconds, and number of features in millions, for all treebanks containing morphological information, with PPL feature included. The highest accuracy for each treebank appears in bold face. Languages appear in decreasing order of average number of morphological attributes per token.

	no-morph			orig			agr			agr+orig		
Lang	UAC	time	feats	UAC	time	feats	UAC	time	feats	UAC	time	feats
hin	89.8	1.4k	1.6	92.5	3.0k	15.5	**93.9**	2.3k	1.6	93.3	3.8k	15.5
hun	88.4	4.4k	5.3	89.0	12.1k	41.6	**90.4**	5.1k	5.3	90.0	13.1k	41.6
ces	84.4	3.4k	4.8	84.6	6.4k	26.6	**86.7**	3.7k	4.8	85.9	7.1k	26.7
tam	79.8	0.3k	0.5	80.6	0.5k	2.0	**83.4**	0.3k	0.5	81.2	0.6k	2.0
slv	82.8	0.6k	1.0	82.1	1.1k	4.7	**83.6**	0.7k	1.0	82.3	1.3k	4.7
dan	88.9	2.4k	1.6	89.2	3.2k	5.8	**90.0**	2.4k	1.6	89.7	3.6k	5.8
lat	63.6	2.2k	1.6	68.8	2.7k	6.7	**73.4**	2.3k	1.7	72.3	2.8k	6.7
nld	88.8	1.9k	3.6	89.5	3.6k	13.4	**90.6**	2.4k	3.6	**90.6**	4.0k	13.4
eus	81.6	0.7k	1.7	82.6	1.2k	5.5	**83.8**	0.8k	1.7	83.6	1.2k	5.5
bul	93.2	1.9k	2.6	93.0	2.6k	8.4	**93.9**	1.8k	2.6	93.5	2.8k	8.4
grc	75.1	9.9k	3.8	79.1	11.9k	15.6	**82.4**	13.3k	3.8	81.4	14.2k	15.6
deu	90.8	0.9k	1.3	91.3	1.3k	3.6	**92.2**	1.0k	1.3	91.9	1.3k	3.6
fin	78.2	0.8k	2.4	79.8	1.3k	8.4	**81.6**	0.9k	2.4	81.3	1.4k	8.4
tur	80.7	1.0k	2.1	82.1	1.8k	5.8	82.2	1.3k	2.1	**82.4**	1.4k	5.8
cat	85.5	2.5k	2.5	84.8	3.0k	6.2	**86.3**	2.2k	2.5	85.6	3.9k	6.2
ara	78.3	6.5k	1.8	78.6	6.5k	3.7	**79.0**	4.5k	1.8	**79.0**	6.0k	3.7
ita	91.1	5.2k	1.8	91.2	5.6k	2.9	**91.5**	4.7k	1.8	**91.5**	5.7k	2.9
por	90.3	8.3k	5.0	90.4	9.1k	7.3	**90.5**	6.4k	5.0	**90.5**	9.1k	7.3
heb	90.6	3.5k	3.1	90.5	4.3k	4.1	**90.8**	3.8k	3.1	90.6	3.4k	4.1
eng	92.7	5.4k	3.1	92.6	4.3k	3.4	**92.8**	5.2k	3.1	**92.8**	4.8k	3.4
cmn	**89.4**	10.4k	6.0	**89.4**	8.8k	6.0	**89.4**	9.2k	6.0	**89.4**	10.4k	6.0

Appendix G

PPL MODIFICATION TO MSTPARSER

The original version of the parser does not contain a feature encapsulating only the head POS, dependent POS, and label of an arc. Experimenting with the `agr` feature set suggested that such a feature might improve performance significantly; I explored this by adding it to the parser. This involved adding code to the `addLabeledFeatures` method in `mstparser/DependencyPipe.java`, as detailed in Listing G.1.

After these lines:

```
add("NTH="+w+" "+wP+suff,fv);
add("NTI="+wP+suff,fv);
add("NTIA="+wPm1+" "+wP+suff,fv);
add("NTIB="+wP+" "+wPp1+suff,fv);
add("NTIC="+wPm1+" "+wP+" "+wPp1+suff,fv);
add("NTJ="+w+suff,fv);
```

Inserted these lines:

```
if (word > 0) {                                                    // if not root
   String head_cpos = instance.cpostags[instance.heads[word]];     // grab head CPOS
   String dep_cpos = instance.cpostags[word];                      // grab dep CPOS
   add("head="+head_cpos+",dep="+dep_cpos+",label="+type, fv);     // add feature
}
```

Listing G.1: Source code for PPL feature.