ACTA UNIVERSITATIS UPSALIENSIS
Studia Linguistica Upsaliensia
14

SICS Dissertation Series
61

Predicting Linguistic Structure with Incomplete and Cross-Lingual Supervision

Oscar Täckström

UPPSALA
UNIVERSITET

Dissertation presented at Uppsala University to be publicly examined in Sal IX, Universitetshuset, Uppsala, Thursday, May 16, 2013 at 10:15 for the degree of Doctor of Philosophy. The examination will be conducted in English.

Abstract
Täckström, O. 2013. Predicting Linguistic Structure with Incomplete and Cross-Lingual Supervision. Acta Universitatis Upsaliensis. *Studia Linguistica Upsaliensia* 14. xii+215 pp. Uppsala. ISBN 978-91-554-8631-0.

Contemporary approaches to natural language processing are predominantly based on statistical machine learning from large amounts of text, which has been manually annotated with the linguistic structure of interest. However, such complete supervision is currently only available for the world's major languages, in a limited number of domains and for a limited range of tasks. As an alternative, this dissertation considers methods for linguistic structure prediction that can make use of incomplete and cross-lingual supervision, with the prospect of making linguistic processing tools more widely available at a lower cost. An overarching theme of this work is the use of structured discriminative latent variable models for learning with indirect and ambiguous supervision; as instantiated, these models admit rich model features while retaining efficient learning and inference properties.

The first contribution to this end is a latent-variable model for fine-grained sentiment analysis with coarse-grained indirect supervision. The second is a model for cross-lingual word-cluster induction and the application thereof to cross-lingual model transfer. The third is a method for adapting multi-source discriminative cross-lingual transfer models to target languages, by means of typologically informed selective parameter sharing. The fourth is an ambiguity-aware self- and ensemble-training algorithm, which is applied to target language adaptation and relexicalization of delexicalized cross-lingual transfer parsers. The fifth is a set of sequence-labeling models that combine constraints at the level of tokens and types, and an instantiation of these models for part-of-speech tagging with incomplete cross-lingual and crowdsourced supervision. In addition to these contributions, comprehensive overviews are provided of structured prediction with no or incomplete supervision, as well as of learning in the multilingual and cross-lingual settings.

Through careful empirical evaluation, it is established that the proposed methods can be used to create substantially more accurate tools for linguistic processing, compared to both unsupervised methods and to recently proposed cross-lingual methods. The empirical support for this claim is particularly strong in the latter case; our models for syntactic dependency parsing and part-of-speech tagging achieve the hitherto best published results for a wide number of target languages, in the setting where no annotated training data is available in the target language.

Keywords: linguistic structure prediction, structured prediction, latent-variable model, semi-supervised learning, multilingual learning, cross-lingual learning, indirect supervision, partial supervision, ambiguous supervision, part-of-speech tagging, dependency parsing, named-entity recognition, sentiment analysis

Oscar Täckström, Uppsala University, Department of Linguistics and Philology, Box 635, SE-751 26 Uppsala, Sweden.

ISSN 1652-1366
ISBN 978-91-554-8631-0
urn:nbn:se:uu:diva-197610 (http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-197610)

SICS Dissertation Series 61
ISSN 1101-1335
ISRN SICS-D-61-SE

Printed by Elanders Sverige AB, 2013

Till Mona-Lisa och Stefan

Acknowledgments

There are many people without whom this dissertation would have looked nothing like it. First and foremost, I want to thank my scientific advisors: Joakim Nivre, Jussi Karlgren and Ryan McDonald; You have complemented each other in the best possible way and by now I see a piece of each one of you in most aspects of my research. Thank you for making my time as a graduate student such a fun and inspiring one. I also want to thank you for your close readings of various drafts of this manuscript — it's final quality owes much to your detailed comments.

Joakim, thank you for sharing your vast knowledge of both computational and theoretical linguistics with me, and for giving me the freedom to follow my interests. As my main supervisor, Joakim had the dubious honor of handling all the bureaucracy involved in my graduate studies; I am very thankful for your swift handling of all that boring, but oh so important, stuff.

Jussi, thank you for all the inspiration, both within and outside of academia, for being such a good friend, and for landing me the job at SICS when I looked to move to Sweden from Canada.

A large part of the work ending up in this dissertation was performed while I was an intern at Google, spending three incredible summers in New York. Many of the contributions in this dissertation were developed in close collaboration with Ryan during this time. Ryan, thank you for being the best host anyone could ask for and for being so supportive of these ideas.

While at Google, I also had the great pleasure to work with Dipanjan Das, Slav Petrov and Jakob Uszkoreit. Parts of this dissertation are based on the papers resulting from our collaboration; I hope that there are more to come!

In addition to my coauthors, I want to thank Keith Hall, Kuzman Ganchev, Yoav Goldberg, Alexander (Sasha) Rush, Isaac Councill, Leonid Velikovich, Hao Zhang, Michael Ringgaard and everyone in the NLP reading group at Google for the many stimulating conversations on natural language processing and machine learning.

The models in chapters 9 and 10 were implemented with an awesome hypergraph-inference library written by Sasha, saving us much time. Thank you for all your help with this implementation. I also want to thank John DeNero and Klaus Macherey, for helping with the bitext extraction and word alignment for the experiments in chapters 8 and 10.

In addition to my advisors, Jörg Tiedemann and Björn Gambäck read and provided detailed comments on earlier versions of this manuscript, for which I am much grateful. Jörg also flawlessly played the role of the public examiner at my final dissertation seminar.

Funding is a necessary evil when you want to do research. Thankfully, my graduate studies were fully funded by the Swedish National Graduate School of Language Technology (GSLT). In addition, GSLT provided a highly stimulating research and study environment.

Parallel to my studies, I have been employed at the Swedish Institute of Computer Science (SICS), under the supervision of Magnus Boman, Björn Levin and, most recently, Daniel Gillblad. Thank you for always supporting my work and for being flexible, in particular during the final stage of this work, when all I could think of was writing, writing, writing. A special thank you goes to the administrative staff at SICS and Uppsala University, for always sorting out any paper work and computer equipment issue.

I also want to thank my colleagues at the Department of Linguistics and Philology at Uppsala University and in the IAM and USE labs at SICS, for providing such a stimulating research and work environment. Fredrik Olsson, Gunnar Eriksson and Magnus Sahlgren, thank you for the many (sometimes heated) conversations on how computational linguistic ought to be done. Kristofer Franzén, thank you for the inspiring musings on food.

My first baby-steps towards academia were taken under the guidance of Viggo Kann and Magnus Rosell. Were it not for their excellent supervision of my Master's thesis work at KTH, which sparked my interest in natural language processing, this dissertation would likely never have been written.

I want to thank the people at DSV / Stockholm University, in particular Hercules Dalianis and Martin Hassel, for stimulating conversations over the past years, and Henrik Boström for giving me the opportunity to give seminars at DSV now and then.

Fabrizio Sebastiani kindly invited me to Pisa for a short research stay at ISTI-CNR. Thank you Fabrizio, Andrea Esuli, Diego Marcheggiani, Giacomo Berardi, Cris Muntean and Diego Ceccarelli, for showing me great hospitality.

Graduate school is not always a walk in the park. The anguish that comes when the results do not is something that only fellow graduate students seem to fully grasp. I especially want to thank Sumithra Velupillai, for our many conversations on research and life in general. I am also very happy to have shared parts of this experience with Baki Cakici, Anni Järvelin, Pedro Sanches, and Olof Görnerup.

Last, but not least, without my dear friends and family, I would never have completed this work. Thank you for staying by my side, even at times when I neglected you in favor of my research. Maria, thank you for the fun times we had in New York! Mom and Dad, thank you for always being there, without ever telling me what to do or where to go.

Oscar Täckström
Stockholm, April 2013

Contents

1 Introduction 1
1.1 Analyzing Language with Statistical Methods 2
1.2 Incomplete and Cross-Lingual Supervision 4
1.3 Contributions 5
1.4 Organization of the Dissertation 6
1.5 Key Publications 8

Part I: Preliminaries 9

2 Linguistic Structure 11
2.1 Structure in Language 11
2.2 Parts of Speech 12
2.3 Syntactic Dependencies 15
2.4 Named Entities 21
2.5 Sentiment 23

3 Structured Prediction 29
3.1 Predicting Structure 29
3.1.1 Characteristics of Structured Prediction 29
3.1.2 Scoring and Inference 31
3.2 Factorization 32
3.2.1 Sequence Labeling 33
3.2.2 Arc-Factored Dependency Parsing 35
3.3 Parameterization 36
3.4 Probabilistic Models 38
3.4.1 Globally Normalized Models 39
3.4.2 Locally Normalized Models 41
3.4.3 Marginalization and Expectation 42
3.5 Inference 43

4 Statistical Machine Learning 45
4.1 Supervised Learning 45
4.1.1 Regularized Empirical Risk Minimization 45
4.1.2 Cost Functions and Evaluation Measures 47
4.1.3 Surrogate Loss Functions 49
4.1.4 Regularizers 52
4.2 Gradient-Based Optimization 53
4.2.1 Online versus Batch Optimization 54

4.2.2 Gradients of Loss Functions and Regularizers 55
4.2.3 Tricks of the Trade 58

Part II: Learning with Incomplete Supervision 61

5 Learning with Incomplete Supervision 63
5.1 Types of Supervision 63
5.2 Structured Latent Variable Models 69
5.2.1 Latent Loss Functions 70
5.2.2 Learning with Latent Variables 73

6 Sentence-Level Sentiment Analysis with Indirect Supervision 77
6.1 A Sentence-Level Sentiment Data Set 78
6.2 Baseline Models 80
6.3 A Discriminative Latent Variable Model 83
6.3.1 Learning and Inference 86
6.3.2 Feature Templates 87
6.4 Experiments with Indirect Supervision 88
6.4.1 Experimental Setup 88
6.4.2 Results and Analysis 89
6.5 Two Semi-Supervised Models 96
6.5.1 A Cascaded Model 96
6.5.2 An Interpolated Model 97
6.6 Experiments with Semi-Supervision 98
6.6.1 Experimental Setup 98
6.6.2 Results and Analysis 98
6.7 Discussion 99

Part III: Learning with Cross-Lingual Supervision 105

7 Learning with Cross-Lingual Supervision 107
7.1 Multilingual Structure Prediction 107
7.1.1 Multilingual Learning Scenarios 108
7.1.2 Arguments For Cross-Lingual Learning 112
7.2 Annotation Projection and Model Transfer 113
7.2.1 Annotation Projection 114
7.2.2 Model Transfer 119
7.2.3 Multi-Source Transfer 120
7.3 Cross-Lingual Evaluation 121

8 Cross-Lingual Word Clusters for Model Transfer 125
8.1 Monolingual Word Clusters 126
8.2 Monolingual Experiments 128
8.2.1 Experimental Setup 128
8.2.2 Cluster-Augmented Feature Models 130

8.2.3 Results ... 131
8.3 Cross-Lingual Word Clusters ... 132
8.3.1 Cluster Projection ... 135
8.3.2 Joint Cross-Lingual Clustering ... 135
8.4 Cross-Lingual Experiments ... 136
8.4.1 Experimental Setup ... 137
8.4.2 Results ... 138

9 Target Language Adaptation of Discriminative Transfer Parsers ... 141
9.1 Multi-Source Delexicalized Transfer ... 141
9.2 Basic Models and Experimental Setup ... 143
9.2.1 Discriminative Graph-Based Parser ... 144
9.2.2 Data Sets and Experimental Setup ... 145
9.2.3 Baseline Models ... 146
9.3 Feature-Based Selective Sharing ... 147
9.3.1 Sharing Based on Typological Features ... 147
9.3.2 Sharing Based on Language Groups ... 149
9.4 Target Language Adaptation ... 150
9.4.1 Ambiguity-Aware Training ... 150
9.4.2 Adaptation Experiments ... 154

10 Token and Type Constraints for Part-of-Speech Tagging ... 157
10.1 Token and Type Constraints ... 158
10.1.1 Token Constraints ... 158
10.1.2 Type Constraints ... 159
10.1.3 Coupled Token and Type Constraints ... 161
10.2 Models with Coupled Constraints ... 163
10.2.1 HMMs with Coupled Constraints ... 164
10.2.2 CRFs with Coupled Constraints ... 165
10.3 Empirical Study ... 166
10.3.1 Experimental Setup ... 166
10.3.2 Type-Constrained Models ... 168
10.3.3 Token-Constrained Models ... 171
10.3.4 Analysis ... 172

Part IV: Conclusion ... 177

11 Conclusion ... 179
11.1 Summary and Main Contributions ... 179
11.2 Future Directions ... 184
11.3 Final Remarks ... 188

References ... 189

Appendix A: Language Codes ... 215

List of Tables

Table 6.1: Number of sentences per document sentiment category 78
Table 6.2: Document- and sentence-level statistics for the labeled test set 79
Table 6.3: Distribution of sentence sentiment per document sentiment 80
Table 6.4: Number of entries per rating in the MPQA polarity lexicon 81
Table 6.5: Sentence sentiment results from document supervision 90
Table 6.6: Sentence-level sentiment results per document category 91
Table 6.7: Sentence-level sentiment accuracy by varying training size 92
Table 6.8: Sentence sentiment results with neutral documents excluded ... 94
Table 6.9: Sentence results for varying numbers of labeled reviews 97
Table 6.10: Sentence sentiment results in the semi-supervised scenario 99
Table 7.1: The universal syntactic dependency rules of Naseem et al. 111
Table 8.1: Additional cluster-based parser features 130
Table 8.2: Cluster-augmented named-entity recognition features 131
Table 8.3: Results of supervised parsing 132
Table 8.4: Results of supervised named-entity recognition 133
Table 8.5: Results of model transfer for dependency parsing 138
Table 8.6: Results of model transfer for named-entity recognition 139
Table 9.1: Typological features from WALS for selective sharing 142
Table 9.2: Values of typological features for the studied languages 143
Table 9.3: Generative versus discriminative models (full supervision) 144
Table 9.4: Results of multi-source transfer for dependency parsing 148
Table 9.5: Results of target language adaptation of multi-source parsers 155
Table 10.1: Tagging accuracies for type-constrained HMM models 168
Table 10.2: Tagging accuracies for token and type constrained models ... 170

List of Figures

Figure 2.1: A sentence annotated with part-of-speech tags 13
Figure 2.2: A sentence annotated with projective syntactic dependencies 16
Figure 2.3: A sentence annotated with non-projective dependencies 17
Figure 2.4: A sentence annotated with named entities 22
Figure 2.5: A review annotated with sentence and document sentiment ... 25
Figure 6.1: Graphical models for joint sentence and document sentiment 84
Figure 6.2: Sentence sentiment precision–recall curves 93
Figure 6.3: Sentence sentiment precision–recall curves (excluding neutral documents) 95
Figure 6.4: Sentence sentiment precision–recall curves (semi-supervised) 100
Figure 6.5: Sentence sentiment precision–recall curves (semi-supervised, observed document label) 101
Figure 7.1: Projection of parts of speech and syntactic dependencies 114
Figure 7.2: Parsing a Greek sentence with a delexicalized English parser 120
Figure 7.3: Different treatments of coordinating conjunctions 122
Figure 8.1: Illustration of cross-lingual word clusters for model transfer 134
Figure 9.1: Arc-factored parser feature templates. 146
Figure 9.2: An example of ambiguity-aware self-training 153
Figure 10.1: Tagging inference space after pruning with type constraints 160
Figure 10.2: Wiktionary and projection dictionary coverage 162
Figure 10.3: Average number of Wiktionary-licensed tags per token 163
Figure 10.4: Relative influence of token and type constraints 172
Figure 10.5: Effect on pruning accuracy from correcting Wiktionary 173
Figure 10.6: Wiktionary pruning mistakes per part-of-speech tag 174

1. Introduction

Language is our most natural and effective tool for expressing our thoughts. Unfortunately, computers are not as comfortable with the natural languages used by humans, preferring instead to communicate in formally specified and unambiguous artificial languages. The goal of *natural language processing* is to change this state of affairs by endowing machines with the ability to analyze and ultimately "understand" human language. Although this may seem very ambitious, the search engines, digital assistants and automatic translation tools that many of us rely on in our daily lives, suggest that we have made at least some headway towards this goal.

Contemporary systems for linguistic processing are predominantly data-driven and based on statistical approaches. That is, rather than being hard-coded by human experts, these systems *learn* how to analyze the linguistic structure underlying natural language from data. However, human guidance and supervision is still necessary for teaching the system how to accurately predict the linguistic structure of interest.[1] This reliance on human expertise forms a major bottleneck in the development of linguistic processing tools. The question studied in this dissertation is therefore central to current research and practice in natural language processing:

> How can partial information be used in the prediction of linguistic structure?

This question is of considerable importance, as a constructive answer would make the development of linguistic processing tools both faster and cheaper, which in turn would enable the use of such tools in a wider variety of applications and languages. While the contributions in this dissertation by no means constitute a complete answer to this question, a variety of modeling approaches and learning methods are introduced that achieve state-of-the-art results for several important applications. The thread shared by these approaches is that they all operate in the setting where only partial information is available to the system. More specifically, the above question is divided into the following two related research questions:

1. How can we learn to make predictions of linguistic structure using incomplete supervision?
2. How can we learn to make predictions of linguistic structure in one language using resources in another language?

[1] In this dissertation, the term *linguistic structure prediction* refers broadly to the automatic attribution of some type of structure to natural language text.

The supervision that can be derived from cross-lingual resources is often incomplete; therefore answering the former question is integral to answering the latter. In the course of this study, we will see that *structured latent variable models*, that is statistical models that incorporate both observed and hidden structure, form a versatile tool, which is well-suited for harnessing diverse sources of incomplete supervision. Furthermore, it will be shown that incomplete supervision, derived from both monolingual and cross-lingual sources, can indeed be used to effectively predict a variety of linguistic structures in a wide range of languages.

In terms of specific applications, incomplete and cross-lingual supervision is leveraged for multilingual *part-of-speech tagging*, *syntactic dependency parsing* and *named-entity recognition*. Furthermore, in the monolingual setting, a method for *fine-grained sentiment analysis* from coarse-grained indirect supervision is introduced. These contributions are spelled out in section 1.3. The remainder of this chapter provides an introduction to the field of natural language processing and statistical machine learning, with a minimum of technical jargon. These subjects are afforded a more rigorous treatment in subsequent chapters.

1.1 Analyzing Language with Statistical Methods

At a high level of abstraction, a linguistic processing system provides a *mapping* that specifies how the linguistic structure underlying natural language text,[2] such as parts of speech, or syntactic and semantic relations, is to be uncovered from its surface form. In the early days of natural language processing, this mapping was composed of hand-crafted rules that specified, for example, how words with particular parts of speech fit together in certain syntactic relations. Instead, modern systems for linguistic analysis typically employ highly complex rules that are automatically induced from data by means of *statistical machine learning* methods.

One reason for relying on statistical learning is that human-curated rule systems quickly become untenable due to the difficulty of manually ensuring the consistency of such systems as the number of rules grow large. Due to the inherent *ambiguity* and *irregularity* of human natural languages, the mapping provided by a high-accuracy linguistic processing system is necessarily tremendously complex. The stride has therefore been towards replacing rule-based systems with statistical ones in the construction of linguistic processing tools. This trend has been especially strong since the early 1990s, spurred by the availability of large data sets and high-power computational resources.[3]

[2] Or speech, albeit this dissertation is focused purely on written text.

[3] The use of statistical methods for linguistic analysis is not new. As early as in the ninth century, Arabic scholars employed crude statistics of letter distributions in cryptanalysis (Al-Kadi, 1992).

While irregularity could in principle be handled by adding more specific rules and by increasing the lexicon, resolving ambiguity typically requires a more global scope on which different rules tend to interact in complex ways. This is because, in order to successfully interpret an utterance, one is required to interpret all of its parts *jointly*; it is not possible to interpret a sentence by considering each of its words in isolation. While this is even more true at the syntactic level (structural ambiguity), where interactions typically have longer range, it is true already at the level of parts of speech (lexical ambiguity).[4]

Consider the task of disambiguating the parts of speech of the following English sentence:

I	saw	her	duck	under	the	table	.
PRON	VERB	PRON	VERB	PREP	DET	NOUN	PUNC
NOUN	NOUN		NOUN	ADJ		VERB	
				ADV			

The potential parts of speech of each word, according to some — necessarily incomplete — lexicon, are listed below each word. Although it is possible for a human to specify rules for how to assign the parts of speech to each word in this example (provided its context), it is very difficult to write maintainable rules that generalize to other utterances and to other contexts. Of course, part-of-speech disambiguation — or *part-of-speech tagging* — is one of the simpler forms of linguistic analysis; the complexity involved in syntactic analysis and semantic interpretation is even more daunting. The last two decades of natural language processing research almost unanimously suggest that statistical learning is better suited to handle this immense complexity.

However, the use of statistical machine learning does not eradicate the need for human labour in the construction of linguistic processing systems. Instead, these methods typically require large amounts of human-curated training data that has been annotated with the linguistic structure of interest, to reach a satisfactory level of performance. For example, a typical *supervised learning* approach to building part-of-speech taggers requires tens of thousands of sentences in which the part of speech of each word has been manually annotated by a human expert. While this is a laborious endeavor, the annotation work required for more complex tasks, such as syntactic parsing, is even more daunting. This is not a factor that has inhibited the construction of linguistic processing tools for the world's major languages too severely. However, the cost of creating the required resources is so high that such tools are currently lacking for most of the world's languages.

The work on *Markov models* and *information theory* by Alan Turing and others during World War II, see MacKay (2003), and of Claude Shannon soon afterwards (Shannon, 1948, 1951), represent two other early key developments towards modern day statistical approaches.

[4]Note that these levels of ambiguity often interact.

1.2 Incomplete and Cross-Lingual Supervision

There are several ways in which knowledge can enter a linguistic processing system. At a high level, we identify the following three sources of knowledge:

1. *Expert rules:* Human experts manually construct rules that define a mapping from input text to linguistic structure. This is typically done in an iterative fashion, in which the mapping is repeatedly evaluated on text data to improve its predictions.
2. *Labeled data:* Human experts — or possibly a crowd of laymen — annotate text with the linguistic structure of interest. A mapping from input text to linguistic structure is then induced by *supervised* machine learning from the resulting labeled data.
3. *Unlabeled data:* Human experts curate an unlabeled data set consisting of raw text and specifies a statistical model that uncovers structure in this data using *unsupervised* machine learning. The inferred structure is hoped to correlate with the desired linguistic structure.

Nothing prevents us from combining these types of knowledge sources. In fact, much research has been devoted to such methods in the last decade within the machine learning and natural language processing communities. The class of combined methods that have received most attention are *semi-supervised* learning methods, which exploit a combination of labeled and unlabeled data to improve prediction. Methods that combine expert rules (or constraints) with unlabeled data are also possible and are sometimes referred to as *weakly supervised*. In these methods, the human-constructed rules are typically used to guide the unsupervised learner towards mappings that are deemed more likely to provide good predictions.

The methods proposed in this dissertation fall under the related concept of learning with *incomplete supervision*. This adds an additional source of knowledge situated between labeled and unlabeled data to the ones above:

4. *Partially labeled data*: Human experts — or possibly a crowd of laymen — annotate text with *some* linguistic structure *related to* the structure that ones wants to predict. This data is then used for *partially supervised* learning with a statistical model that exploits the annotated structure to infer the linguistic structure of interest.

Several different sources of incomplete supervision will be explored in this dissertation. In particular, we will consider learning with *cross-lingual* supervision, where (possibly incomplete) annotation in a source language is leveraged to infer the linguistic structure of interest in a target language.

To exemplify, in chapter 10, we show that it is possible to construct accurate and robust part-of-speech taggers for a wide range of languages, by combining (1) manually annotated resources in English, or some other language for which such resources are already available, with (2) a *crowd-sourced* target-language specific lexicon, which lists the potential parts of speech that

each word may take in some context, at least for a subset of the words. Both (1) and (2) only provide partial information for the part-of-speech tagging task. However, taken together they turn out to provide substantially more information than either taken alone. While the source and type of partial information naturally varies between tasks, our methods are grounded in a general class of discriminative probabilistic models with constrained latent variables. This allows us to make use of well-known, efficient and effective, methods for inference and learning, freeing up resources to focus on modeling aspects and on assembling problem-specific knowledge-sources.

Besides a purely scientific and engineering interest, our interest in learning with incomplete supervision is a pragmatic one, motivated by the inherent trade-off between prediction performance and development cost. Fully labeled data is typically costly and time-consuming to produce and requires specialist expertise, but when available typically allows more accurate prediction. Unlabeled data, on the other hand, is often available at practically zero marginal cost, but even when fed with massive amounts of data, unsupervised methods can typically not compete with fully supervised methods in terms of prediction performance. Partially labeled data allow us to strike a balance between annotation cost and prediction accuracy.

1.3 Contributions

The contributions in this dissertation are all related to the use of partial information in the prediction of linguistic structure. Specifically, we contribute to the understanding of this topic along the following dimensions:

1. We propose a structured discriminative model with latent variables, enabling *sentence-level sentiment* to be inferred from *document-level sentiment* (review ratings). While other researchers have improved on our results since the original publication of this work, the models presented in this dissertation still represent a competitive baseline in this setting.
2. We introduce the idea of *cross-lingual word clusters*, that is, morphosyntactically and semantically informed groupings of words, such that the groupings are consistent across languages. A simple algorithm is proposed for inducing such clusters and it is shown that the resulting clusters can be used as a vehicle for transferring linguistic processing tools from resource-rich source languages to resource-poor target languages.
3. We show how *selective parameter sharing*, recently proposed by Naseem et al. (2012), can be applied to discriminative graph-based dependency parsers to improve the transfer of parsers from multiple resource-rich source languages to a resource-poor target language. This yields the best published results for multi-source syntactic transfer parsing to date.
4. We introduce an *ambiguity-aware* training method for target language adaptation of structured discriminative models, which is able to lever-

age automatically inferred ambiguous predictions on unlabeled target language text. This brings further improvements to the model with selective parameter sharing in item 3.

5. We introduce the use of coupled *token and type constraints* for part-of-speech tagging. By combining type constraints derived from a crowdsourced tag lexicon with token constraints derived via cross-lingual supervision, we achieve the best published results to date in the scenario where no fully labeled resources are available in the target language.

In addition to these contributions, we give comprehensive overviews of approaches to learning with no or incomplete supervision and of multilingual and, in particular, cross-lingual learning.

1.4 Organization of the Dissertation

The remainder of this dissertation is organized into the following four parts:

Part I: Preliminaries

Chapter 2 provides an introduction to the various types of linguistic structure that play a key part in the remainder of the dissertation: *parts of speech*, *syntactic dependencies*, *named entities*, and *sentiment*. These structures are described and motivated from a linguistic as well as from a practical perspective and a brief survey of computational approaches is given.

Chapter 3 introduces the framework of structured prediction on which all of our methods are based. We describe how structured inputs and outputs are represented and scored in a way that allows for efficient inference and learning, and we discuss probabilistic models. In particular, we show how different linguistic structures are represented in this framework.

Chapter 4 provides an introduction to statistical machine learning for structured prediction, with a focus on supervised methods. Different evaluation measures are described and an exposition of regularized empirical risk minimization with different loss functions is given. Following this is a brief discussion of methods for gradient-based optimization, and some tricks of the trade for implementing these methods efficiently.

Part II: Learning with Incomplete Supervision

Chapter 5 describes various ways to learn from no or incomplete supervision, including unsupervised, semi-supervised and, in particular, structured latent variable models. These key tools are described in general terms in this chapter, while concrete instantiations are developed in chapters 6 and 8 to 10. We further discuss related work, such as constraint-driven learning and posterior regularization.

Chapter 6 proposes the use of structured probabilistic models with latent variables for sentence-level sentiment analysis with document-level supervision. An extensive empirical study of an indirectly supervised and a semi-supervised variant of this model is provided, in the common scenario where document-level supervision is available in the form of product review ratings. Additionally, the manually annotated test set, which is used for evaluation, is described.

Part III: Learning with Cross-Lingual Supervision

Chapter 7 gives an overview of multilingual linguistic structure prediction, with a focus on cross-lingual learning for part-of-speech tagging, syntactic dependency parsing and named-entity recognition. Specifically, methods for annotation projection with word-aligned bitext and direct model transfer by means of cross-lingual features are discussed. Again, these methods are described in general terms, while our contributed methods to this area are found in chapters 8 to 10.

Chapter 8 introduces the idea of cross-lingual word clusters for cross-lingual transfer of models for linguistic structure prediction. First, monolingual word clusters are evaluated for use in semi-supervised learning. Second, an algorithm for cross-lingual word cluster induction is provided. Both types of clusters are evaluated for syntactic dependency parsing, as well as for named-entity recognition, across a variety of languages.

Chapter 9 studies multi-source discriminative transfer parsing by means of selective parameter sharing, based on typological and language-family characteristics. First, different ways of selective parameter sharing in a discriminative graph-based dependency parser are described. Second, the idea of ambiguity-aware self- and ensemble-training of structured probabilistic models is introduced and applied to the selective sharing model.

Chapter 10 considers the construction of part-of-speech taggers for resource-poor languages, by means of constraints defined at the level of both tokens and types. Specifically, coupled token and type constraints provide an ambiguous signal, which is used to train both generative and discriminative sequence-labeling models. The chapter ends with an empirical study and a detailed error analysis, where the relative contributions of type and token constraints are compared.

Part IV: Conclusion

Chapter 11 summarizes the dissertation and its main contributions. Finally, we conclude with an outline of pertinent directions for future work.

1.5 Key Publications

The material in this dissertation — in particular the material in chapters 6 and 8 to 10 — is to a large extent based on the following publications:

Oscar Täckström and Ryan McDonald (2011a).[5] Discovering Fine-Grained Sentiment with Latent Variable Structured Prediction Models. In *Proceedings of the European Conference on Information Retrieval (ECIR)*, pages 368–374, Dublin, Ireland.

Oscar Täckström and Ryan McDonald (2011c). Semi-Supervised Latent Variable Models for Sentence-Level Sentiment Analysis. In *Proceedings of the Conference of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT): Short papers*, pages 569–574, Portland, Oregon, USA.

Oscar Täckström, Ryan McDonald and Jakob Uszkoreit (2012). Cross-Lingual Word Clusters for Direct Transfer of Linguistic Structure. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 477–487, Montreal, Canada.

Oscar Täckström (2012). Nudging the Envelope of Direct Transfer Methods for Multilingual Named Entity Recognition. In *Proceedings of the NAACL-HLT Workshop on Inducing Linguistic Structure (WILS)*, pages 55–63, Montreal, Canada.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald and Joakim Nivre (2013). Token and Type Constraints for Cross-Lingual Part-of-Speech Tagging. *Transactions of the Association for Computational Linguistics*, 1, pages 1–12. Association for Computational Linguistics.

Oscar Täckström, Ryan McDonald and Joakim Nivre (2013). Target Language Adaptation of Discriminative Transfer Parsers. Accepted for publication in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Atlanta, Georgia, USA.

[5] This paper is also available as an extended technical report (Täckström and McDonald, 2011b), on which much of the material in chapter 6 is based.

Part I:
Preliminaries

2. Linguistic Structure

This chapter introduces the four types of linguistic structure that are considered in the dissertation: parts of speech, syntactic dependencies, named entities, and sentence- and document-level sentiment. The discussion is kept brief and non-technical. Technical details on how these structures can be represented, scored and inferred are given in chapter 3, while methods for learning to predict these structures from fully annotated data are presented in chapter 4. Chapter 5 discusses methods for learning to predict these structures from incomplete supervision, while chapter 7 discusses learning and prediction in the multilingual and, in particular, learning with cross-lingual supervision. Before introducing the linguistic structures, we provide a brief characterization of natural language processing and what we mean by linguistic structure.

2.1 Structure in Language

As a field, natural language processing is perhaps best characterized as a diverse collection of tasks and methods related to the analysis of written human languages. For scientific, practical and historical reasons, spoken language has mostly been studied in the related, but partly overlapping, field of *speech processing* (Holmes and Holmes, 2002; Jurafsky and Martin, 2009).

A canonical linguistic processing system takes some textual representation, perhaps together with additional metadata, as its input and returns a structured analysis of the text as its output. The output can be a formal representation, such as in syntactic parsing. In other cases, both the input and the output is in the form of natural language text, such as in machine translation and automatic summarization. In some cases, such as in natural language generation, the input may be some formal representation, whereas the output is in natural language. Following Smith (2011), we use the term *linguistic structure* to collectively refer to any structure that the system is set out to infer, even in cases when the structure in question may not be of the kind traditionally studied by linguists.

Most work in natural language processing simply treat a text as a sequence of symbols, although other dimensions, such as paragraph structure, layout and typography, may be informative for some tasks. Throughout, we will assume that the input text has been split into sentences, which have further

been tokenized into words. However, we want to stress that in some languages, such as Chinese, segmenting a sequence of characters into words is not at all trivial. In fact, the problem of *word segmentation* in these languages is the subject of active research (Sproat et al., 1996; Maosong et al., 1998; Peng et al., 2004). *Sentence-boundary detection* is also a non-trivial problem (Reynar and Ratnaparkhi, 1997; Choi, 2000), in particular in less structured text genres, such as blogs and micro-blogs, where tokenization can also be highly problematic (Gimpel et al., 2011). In Thai, this is a notoriously difficult problem for any type of text (Mittrapiyanuruk and Sornlertlamvanich, 2000).

Of the four types of linguistic structure considered in this dissertation, *parts of speech* and *syntactic dependencies* stand out in that they have a rich heritage in syntactic theory and have been studied by grammarians at least since antiquity (Robins, 1967). Broadly taken, the term *syntax* refers to "the structure of phrases and sentences" (Kroeger, 2005, p. 26), or to the "principles governing the arrangement of words in a language" (Van Valin, 2001, p. 1). There are many theories and frameworks for describing and explaining syntactic phenomena; a goal shared by most is to provide a compact abstract description of a language that can still explain the surface realization of that language (Jurafsky and Martin, 2009). Almost all syntactic theories take the clause or the sentence as their basic object of study. The study of word structure and word formation is known as *morphology* (Spencer and Zwicky, 2001), whereas sentence organization and inter-sentential relations is the domain of *discourse* (Marcu, 2000; Gee, 2011). Finally, *semantics* is the study of *meaning* as conveyed in text (Lappin, 1997). While computational approaches to morphology, discourse and semantics are all important parts of natural language processing, we will not study these topics further. However, it is our belief that most of the methods developed in this dissertation may be applicable to the automatic processing of these aspects as well.

In addition to parts of speech and syntactic dependencies, we will also consider *named entities*. While not strictly part of a syntactic analysis, named entities are defined at the word/phrase level and closely related to syntactic structure, whereas the final type of structure, *sentiment*, will here be studied at the level of full sentences and documents. Nevertheless, syntactic structure plays an important role in more fine-grained approaches to sentiment analysis.

2.2 Parts of Speech

While frameworks for syntactic analysis often differ substantially in their repertoire of theoretical constructs, most acknowledge the categorization of words (*lexical items*) into *parts of speech*.[1] This is an important concept in

[1]Other terms for this concept include *word class* and *syntactic category*. According to Van Valin (2001), the term *lexical category* is preferred by most contemporary linguists. Nevertheless, we

John	quickly	handed	Maria	the	red	book	.
NOUN	ADV	VERB	NOUN	DET	ADJ	NOUN	PUNC

Figure 2.1. A sentence annotated with coarse-grained part-of-speech tags from the tag set in Petrov et al. (2012).

theoretical linguistics, as parts of speech play a fundamental role, for example, in morphological and syntactic descriptions (Haspelmath, 2001). In linguistic processing applications, parts of speech are rarely of interest in themselves, but they are still highly useful, as they are often relied upon for accomplishing higher-level tasks, such as syntactic parsing, named-entity recognition and machine translation. Figure 2.1 shows an example sentence, where each word has been annotated with its part of speech. The particular set of part-of-speech tags shown in this example are taken from the "universal" coarse-grained tag set defined by Petrov et al. (2012).

Characterization

In traditional school grammar, the parts of speech are often defined semantically, such that nouns are defined as denoting *things, persons and places*, verbs as denoting *actions and processes* and adjectives as denoting *properties and attributes* (Haspelmath, 2001). However, using semantic criteria to define parts of speech is problematic. When constrained to a single language, these definitions may be appropriate for a set of prototypical words/concepts (Croft, 1991), but there are many words that we undisputedly want to place in these categories that do not fit such semantic criteria. Instead, words are delineated into parts of speech based primarily on morphosyntactic properties, while semantic criteria are used to name the resulting lexical categories by looking at salient semantic properties of the words in each induced category (Haspelmath, 2001). For example, a category whose prototypical members are primarily words denoting things, is assigned the label *nouns*. Further criteria may be involved; often a combination of semantic, pragmatic and formal (that is, morphosyntactic) criteria are employed (Bisang, 2010). Since there is large morphosyntactic variability between languages (Dryer and Haspelmath, 2011), it follows that any grouping of lexical items by their parts of speech must be more or less language specific. Moreover, only nouns, verbs and, to some degree, adjectives, are generally regarded by linguists to be universally available across the world's languages (Croft, 1991; Haspelmath, 2001). For the most part, we will consider the palette of parts of speech as fixed and given, so that our task is to learn how to automatically tag each word in a text with its correct part of speech. This task is referred to as *part-of-speech tagging*. However, we will briefly return to the issue of linguistic

have decided to use the term part of speech, as this is the convention in the computational linguistics community.

universality when discussing cross-lingual prediction of linguistic structure in chapter 7.

Computational approaches

The first working computational approach to part-of-speech tagging was the TAGGIT system by Greene and Rubin (1971). This was a rule-based system, whose functioning was determined by hand-crafted rules. Contemporary systems for part-of-speech tagging are instead based almost exclusively on statistical approaches, as pioneered — independently, it seems — by Derouault and Merialdo (1986), Garside et al. (1987), DeRose (1988) and Church (1988). These were all supervised systems, based on Hidden Markov Models (HMMs), automatically induced from labeled corpora. Since this initial work, a wide variety of approaches have been proposed to this task. For example, in the 1990s, Brill (1992, 1995) combined the merits of rule-based and statistical approaches in his framework of *transformation-based learning* and Ratnaparkhi (1996) pioneered the use of *maximum entropy* models (Berger et al., 1996), while Daelemans et al. (1996) popularized the use of *memory-based learning* (Daelemans and van den Bosch, 2005) for this and other tasks. Brants (2000) returned to the HMM framework with the TNT tagger, based on a second-order HMM. The TNT tagger is still in popular use today, thanks to its efficiency and robustness. Currently, part-of-speech tagging is commonly approached with variants of *conditional random fields* (CRFs; Lafferty et al., 2001). As discussed in section 3.4, HMMs and CRFs are similar probabilistic models that differ mainly in their model space and in their statistical independence assumptions.

According to Manning (2011), contingent on the availability of a sufficient amount of labeled training data, supervised part-of-speech taggers for English now perform almost at an accuracy of 97%, which is claimed to be very close to human-level performance. Manning argues that the remaining gap to human-level performance should be addressed by improving the gold standard corpus annotations, rather than by improving tagging methods. Similar points were raised by Källgren (1996), who argued for the use of underspecified tags in cases where human annotators cannot agree on a single interpretation of an ambiguous sentence. At the time of writing, labeled corpora for training supervised part-of-speech taggers are available for more than 20 languages with average supervised accuracies in the range of 95% (Petrov et al., 2012). However, Manning also points out that these results only hold for the *in-domain* setting, where the data used for both training and evaluating the system belong to the same domain. When moving to other domains, for example, when training the system on edited news text and then applying it to general non-editorial text, there is typically a substantial drop in accuracy for part-of-speech tagging systems; a drop in accuracy from above 95% down to around 90% is not uncommon (Blitzer et al., 2006). This is a more general

problem for any natural language processing system and *domain adaptation* is therefore a topic of active research.

2.3 Syntactic Dependencies

Syntax is at the heart of formal linguistics and while typically not an end goal in linguistic processing, automatic syntactic analysis — *syntactic parsing* — is fundamental to many down-stream tasks such as machine-translation (Chiang, 2005; Collins et al., 2005; Katz-Brown et al., 2011), relation-extraction (Fundel et al., 2007) and sentiment analysis (Nakagawa et al., 2010; Councill et al., 2010). Contemporary approaches to syntax, in formal as well as in computational linguistics, are dominated by two frameworks, which are based on the notion of *constituency* and of *dependency*, respectively.

For the most part of the last century, in particular in the Anglo-American tradition, most linguists have focused on constituency grammars. Similar to the morphosyntactic definition of parts of speech, a *constituent* can be loosely defined as a grouping of words based on the distributional behavior of the group, in particular with respect to word order (Kroeger, 2005). The noun phrase (NP) is a canonical type of constituent, which in English can be partly characterized by its tendency to directly precede verbs. For example, in each of *[the fox] jumps*; *[the quick fox] jumps*; and *[the quick brown fox] jumps*, the NP (in brackets) forms a unit with this characteristic, while only the head noun (*fox*) of the NP shares this trait. In most constituency-based syntactic formalisms, constituents are restricted to form contiguous spans. For languages with strict word order, such as English, this is not a severe restriction, whereas languages with freer word order, such as Czech, Russian, or Finnish, are more difficult to describe in terms of constituents with this restriction (Nivre, 2006). In dependency grammars, which is the syntactic framework considered in this dissertation, the primary notion is instead that of a binary (asymmetric) *dependency relation* between two words, such that one word is designated as the *head* and the other word is designated as the *dependent*, with the dependent being considered to be subordinate to its head (Nivre, 2006). These relations are often visualized as directed arcs between words, as shown in the example in fig. 2.2. The direction of the arcs reflects the asymmetry of the relation; a common convention, adhered to here, is that arcs are defined as pointing from the head word to its dependent(s). See de Marneffe and Manning (2008) for a description of the dependency types (the labels above the arcs) used in figs. 2.2 and 2.3.

Characterization

Nivre (2006) summarizes a list of commonly used criteria for establishing the distinction between head and dependent in a construction. As with parts of speech, it is difficult to fully establish this distinction by formal criteria in

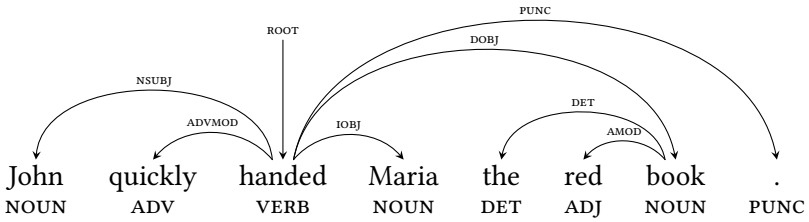

Figure 2.2. A sentence annotated with *projective* syntactic dependencies.

a way that covers every conceivable case, but these criteria are commonly employed. In summary, the head may be loosely defined as the word that determines the syntactic and semantic category of a construction; the head is a word that may replace the construction, while the dependents provide a specification for, or a refinement of, the construction; and finally the head is a word that determines the form and position of its dependents. Similarly, Van Valin (2001, p. 101), as several authors before him, makes a distinction between three types of syntactic dependencies. He defines a *bilateral* dependence as a relation where "neither the head nor the dependent(s) can occur without the other(s)"; a *unilateral* dependence as one in which "the head can occur without dependents in a particular type of construction, but the dependents cannot occur without the head", whereas a *coordinate* dependence, finally, is treated as a special type of dependence between two *co-heads*. The proper way of analyzing coordination in terms of dependency is a much debated topic and there are several competing ways of analyzing such constructions (Nivre, 2006). Some give coordination a different status from dependence (or *subordination*) altogether, rather than shoehorning coordination into a dependency relation (Tesniére, 1959; Hudson, 1984; Kahane, 1997). These decisions are particularly problematic when we consider multilingual syntactic analysis, where the use of different criteria in the creation of manually annotated treebanks makes cross-lingual transfer, comparison and evaluation difficult. See chapter 7 for further discussion of these issues.

An important concept in dependency grammars is that of *valency* (Tesniére, 1959),[2] which captures how the head word, in particular verbs, constrain the morphosyntactic properties of their dependent(s). In addition, the semantic interpretation of a verb places constraints on the number of *semantic roles* that needs to be filled and the form of the dependent(s) that fill those role(s). For example, the English transitive verb *give* is characterized by the requirement of having a subject as well as a direct and an indirect object, with constraints such that these words need to be placed in a certain order and that the subject and the indirect object generally needs to be animate enti-

[2]This is similar to the concept of *subcategorization* in constituency theory (Sag et al., 2003).

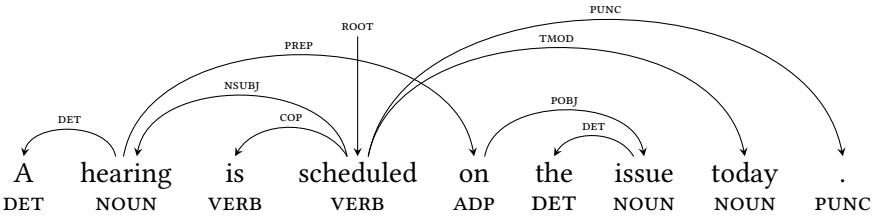

Figure 2.3. A sentence annotated with *non-projective* syntactic dependencies.

ties. There are further connections between syntactic and semantic relations. These connections have been exploited, for example, by Johansson (2008) and Das (2012) in computational approaches to frame-semantic parsing (Fillmore, 1982).

Most dependency syntactic formalisms subscribed to in computational linguistics enforce the fundamental constraint that the set of directed dependencies (arcs) for a given sentence must form a connected directed rooted tree, such that each word has a single head (Nivre, 2006).[3] In graph-theoretic terminology, a directed graph with this property is known as an *arborescence* (Tutte, 2001). The head of the sentence, typically the finite verb, is represented by letting it be the dependent of an artificial *root* word, as shown in figs. 2.2 and 2.3

An important distinction is that between *projective* and *non-projective* dependencies. Informally, a dependency tree for a sentence is projective only if the dependencies between all its words — placed in their natural linear order — can be drawn in the plane above the words, such that no arcs cross and no arc spans the root of the tree. Equivalently, every subtree of a projective dependency tree is constrained to have a contiguous yield. The dependency tree in fig. 2.2 is projective (not the non-crossing nested arcs), while fig. 2.3 shows a non-projective dependency tree (note the crossing arcs). Languages with rigid word order tend to be analyzed as having mostly projective trees, while languages with more free word order tend to exhibit non-projectivity to a higher degree (McDonald, 2006). Yet, even languages that give rise to non-projective dependency analyses tend to be only mildly non-projective (Kuhlmann, 2013). Though different in nature, (contiguous) constituent grammars can often be converted to (projective) dependency structures by means of a relatively small set of *head percolation* rules (Magerman, 1995; Collins, 1997). In fact, treebanks annotated with dependency structure are often produced by an automatic conversion of constituency structure based on such head percolation rules (Yamada and Matsumoto, 2003).

[3] Albeit true of many dependency syntactic formalisms, this constraint is by no means subscribed to by all dependency-based syntactic theories. See, for example, the *word grammar* of Hudson (1984) and the *meaning-text theory* of Mel'čuk (1988).

In the remainder, we will largely ignore linguistic issues and instead focus on the problem of dependency parsing, assuming a given dependency syntactic formalism as manifested in a manually annotated treebank. Furthermore, we restrict ourselves to the case of projective dependencies. However, we will briefly return to some of these issues when discussing cross-linguistic issues in dependency parsing in chapter 7.

Computational approaches

Just as early approaches to part-of-speech tagging were rule-based, early approaches to dependency parsing were based on hand-crafted grammars, for example, Tapanainen and Järvinen (1997). Such approaches are still in active use today. However, contemporary research on, and practical use of, dependency parsing is completely dominated by data-driven approaches. In early work on data-driven dependency parsing, Collins et al. (1999) exploited the fact that a constituency tree can be converted into a projective dependency tree, in order to use a constituency parser to predict dependency structure. However, since constituency parsers are typically substantially more computationally demanding than dependency parsers, later research focused on native approaches to dependency parsing. These can largely be grouped into the two categories of *graph-based* and *transition-based* parsing (McDonald and Nivre, 2007), which differ mainly in the ways in which they decompose a dependency tree when computing its score.

Graph-based parsers decompose the dependency tree either into individual arcs that are scored separately (Eisner, 1996; Ribarov, 2004; McDonald et al., 2005; Finkel et al., 2008), or into higher-order factors in which several arcs are treated as a unit with respect to scoring (McDonald and Pereira, 2006; Carreras, 2007; Smith and Eisner, 2008; Koo and Collins, 2010; Zhang and McDonald, 2012). These units are assembled into a global solution, with the constraint that the result is a valid dependency tree. A related class of parsers are based on integer linear programming (ILP; Schrijver, 1998), in which interactions between arcs of arbitrary order can be incorporated (Riedel and Clarke, 2006; Martins et al., 2009). While higher-order models can often yield better results compared to lower-order models, and in particular compared to first-order (arc-factored) models, this comes at a substantial increase in computational cost. This is particularly true of models based on ILP, for which inference is NP-hard in general. Much research effort has therefore been devoted to approximate methods that can reduce the computational cost, hopefully at only a small cost in accuracy. This include coarse-to-fine methods, such as structured prediction cascades (Weiss and Taskar, 2010), where efficient lower-order models are used to filter the hypotheses that are processed with a higher-level model (Rush and Petrov, 2012). Another approach is that of Smith and Eisner (2008), who cast the problem as a graphical model (Wainwright and Jordan, 2008), in which approximate inference is performed by belief propagation. In order to speed up ILP-based models,

Martins et al. (2009) proposed to use a linear programming (LP) relaxation as an approximation to the ILP. Variational inference (Martins et al., 2010) and delayed column-generation (Riedel et al., 2012) are other recently proposed techniques for speeding up inference with LP relaxations. Another recent approximate inference method for higher-order models is that of Zhang and McDonald (2012), who adopt the idea of cube-pruning, originally introduced in the machine translation community (Chiang, 2007). Finally, dual decomposition has recently gained popularity as a technique for performing inference in higher-order models by combining two or more tractable subproblems via Lagrangian relaxation (Koo et al., 2010; Martins et al., 2011).

One dimension of dependency grammar that profoundly constrains parsing models is that of projectivity. When the dependencies are all assumed to be projective, most graph-based models use variants of the chart-parsing algorithm of Eisner (1996), a bottom-up dynamic programming algorithm, similar to early algorithms of Hays (1964) and Gaifman (1965). In the non-projective case, on the other hand, inference corresponds to the problem of finding a maximum spanning tree (McDonald et al., 2005). This inference problem can be solved exactly for arc-factored models, where the spanning tree is composed directly from individual arcs, but was shown to be NP-hard for higher-order models by McDonald (2006); see also the related result of Neuhaus and Bröker (1997). For the case of second-order non-projective models, an approximate method was proposed by McDonald and Pereira (2006).

As discussed, in graph-based parsers a dependency tree is decomposed into smaller parts that are scored individually and then combined in a way that is (approximately) globally optimal. In transition-based parsing, on the other hand, the dependency tree is instead built up step-by-step by the iterative application of a small set of *parser actions*, where the action at each step is predicted by a classifier trained with some machine learning method. Although many different *transition systems* have been proposed, they most commonly correspond to a shift-reduce style algorithm, where the sentence is traversed in some pre-specified order, such as left-to-right. An early algorithm in this vein is that of Covington (2001), which stands out in the transition-based parsing literature in that it can directly handle non-projective dependencies. Related shift-reduce style algorithms were given by Yamada and Matsumoto (2003) and Nivre (2003). By employing a head-directed *arc-eager* strategy, the latter is able to build a projective dependency tree in time linear in the sentence length. Nivre and Nilsson (2005) later extended this method to the case of non-projective dependencies by a *pseudo-projective* tree transformation, which annotates a projective tree with information that can be used to recover non-projective dependencies in a post-processing step.

These early approaches to transition-based parsing were all based on a *greedy* search strategy, in which only one action is taken at each step and where there is no possibility to reconsider past actions. This makes these approaches brittle and prone to error propagation, where an early mistake

has a negative influence on future actions. The most popular solution to this problem is to use beam search, in which k hypotheses (partially constructed trees) are simultaneously explored (Duan et al., 2007; Huang, 2008b; Zhang and Clark, 2008; Zhang and Nivre, 2011; Huang et al., 2012a).[4] While beam search, like greedy search, is an inexact search strategy, Huang and Sagae (2010) showed that transition-based parsing can also be cast as dynamic programming by a clever state-merging technique. These results where recently generalized by Kuhlmann et al. (2011). However, because beam search is so straightforward to implement, it still dominates in practical use. It should also be pointed out that in order to use tractable dynamic programming for transition-based parsing, only quite crude features can be used. This is a drawback, as rich features has been shown to be necessary for state-of-the-art results with transition-based methods (Zhang and Nivre, 2011).

While scoring is an integral part of graph-based models (most approaches use (log-)linear score functions, described in section 3.1), the classifier used in transition-based parsers is more or less independent of the transition system employed. Many methods have been proposed to learn the action classifier used with greedy transition-based parsing. For example, Kudo and Matsumoto (2000) and Yamada and Matsumoto (2003) used support vector machines (SVMs; Cortes and Vapnik, 1995), Nivre (2003) used memory-based learning (Daelemans and van den Bosch, 2005), while Attardi (2006) used a maximum-entropy (log-linear) model (Berger et al., 1996). When using beam search, the *structured perceptron* (Collins, 2002) is the dominating learning algorithm. Other learning algorithms that have been applied in this scenario are *learning as search optimization* (LaSO; Daumé and Marcu, 2005), *search-based structured prediction* (Searn; Daumé III et al., 2009) and the structured perceptron with inexact search (Huang et al., 2012b).

Although graph-based and transition-based parsers may seem quite different, their merits have been combined (Zhang and Clark, 2008; Zhang and Nivre, 2011). This can be beneficial, as they have been shown to make slightly different types of errors (McDonald and Nivre, 2007).

The more complex methods described above have provided significant improvements in supervised dependency parsing. However, in the cross-lingual projection and multilingual selective sharing scenarios that we consider in this dissertation, the state of the art is still performing at a level substantially below a supervised arc-factored model. We therefore restrict our attention to arc-factored models with exact inference and transition-based models with beam search inference. However, as the performance is raised in these scenarios, we may need to consider more complex models.

[4] At each step all transitions from the current k hypotheses are scored, whereafter the k most likely hypotheses are kept for the next step.

2.4 Named Entities

There are many ways in which a physical, or abstract, entity may be referred to in text. For example, *Stockholm*, *Sthlm* and *the capital of Sweden* all denote the same physical place. In the first two of these expressions, the entity is referred to by a *proper name* and we say that the entity in question is a *named entity*. The automatic recognition and classification of such entities in text is known as *named-entity recognition* (Nadeau and Sekine, 2007).

Initially, named-entity recognition was construed as a subtask of the more ambitious task of *information extraction* (Cowie and Wilks, 2000), which in addition to extracting and classifying named entities, aims at disambiguating the entities and to find interesting relations between them. However, the need for named-entity recognition also arises in applications where the entities themselves are first class objects of interest, such as in *Wikification* of documents (Ratinov et al., 2011), in which entities are linked to their Wikipedia-page,[5] and in applications where knowledge of named entities is not an end-goal in itself, but where the identification of such entities can boost performance, such as machine translation (Babych and Hartley, 2003) and question answering (Leidner et al., 2003). For this reason, named-entity recognition is an important task in its own right. The advent of massive machine readable factual databases, such as Freebase and Wikidata,[6] will likely push the need for automatic extraction tools further. While these databases store information about entities and relationships between entities, recognition of these entities *in context* is still a non-trivial problem. As an example, *Jobs* may be a named entity in some context, such as in *Jobs created Apple*, while not in others, such as in *Jobs created by Microsoft*.

A related driving force behind research on named-entity recognition has been the idea of the *semantic web* (Tim Berners-Lee and Lassila, 2001), which as argued by Wilks and Brewster (2009) seems difficult, if not impossible, to realize without the help of automatic tools. In this vein, "Web-scale" named-entity recognition was proposed and explored by Whitelaw et al. (2008).

Characterization

While early research on recognition of named entities focused exclusively on the recognition and classification of proper names, interest was quickly expanded to the recognition of time expressions and expressions of numerical quantities, such as money (Nadeau and Sekine, 2007).[7] In terms of entity type categorization, the field has since then retained a quite narrow focus on a small number of fundamental entity types. A typical categorization can be found in that defined by the Message Understanding Conferences (MUC;

[5] In many cases, this also requires that entities are disambiguated, so that the correct Wikipedia page can be linked to.

[6] See http://www.freebase.com/ and http://www.wikidata.org/ — February 4, 2013.

[7] This section is largely based on the survey of Nadeau and Sekine (2007).

[Steve Jobs]$_{\text{PER}}$ created [Apple]$_{\text{ORG}}$ in [Silicon Valley]$_{\text{LOC}}$.

Steve	Jobs	created	Apple	in	Silicon	Valley	.
B-PER	I-PER	O	B-ORG	O	B-LOC	I-LOC	O

Figure 2.4. A sentence annotated with named entities. **Top**: entities annotated as bracketed chunks. **Bottom**: the same entities annotated with the BIO-encoding.

Grishman and Sundheim, 1996), according to which entities are grouped into *persons*, *organizations* and *locations*. This categorization was also used by the CoNLL shared tasks on multilingual named-entity recognition (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003), where an additional *miscellaneous* category was introduced, reserved for all entities that cannot be mapped to any of the former three categories. A notable exception to these coarse-grained categorizations can be found in the taxonomy proposed by Sekine and Nobata (2004) specifically for news text, which defines a hierarchy of named entities with over 200 entity types. Another research direction that has led to more fine-grained entity categories can be found within bioinformatics, in particular as embodied in the GENIA corpus (Ohta et al., 2002). However, most work on named-entity recognition for general text is still based on very coarse-grained categorizations; for practical reasons, this is true of the present dissertation as well.

Typically, in order to simplify their definition and extraction, named entities are restricted to form non-overlapping chunks of contiguous tokens, such that each chunk is assigned an entity category. There are many different ways to encode such non-overlapping chunks. The most used encoding is the *begin-inside-outside* (BIO) encoding (Ramshaw and Marcus, 1995), in which each lexical item is marked as constituting the beginning of a chunk (B-X), as belonging to the inside of a chunk (I-X) or as being outside of any chunk (O). Here, X is the category of the entity which the chunk denotes, for example, PER (person), LOC (location), ORG (organization) or MISC (miscellaneous). Figure 2.4 shows an example sentence, where each named entity has been marked and categorized in this manner. Other encodings have been proposed as well, as discussed by Ratinov and Roth (2009).

Computational approaches

Not surprisingly, early work on information extraction and named-entity recognition was predominantly focused on methods based on rule-based pattern matching (Andersen et al., 1986) and custom grammars (Wakao et al., 1996). Such rule-based systems are still in active use today, a prominent example being the recently released multilingual system by Steinberger et al. (2011), which relies on high-precision rules that need to be hand-crafted separately for each language. These approaches tend to provide high precision

at the expense of recall; see section 4.1.2 for definitions of precision and recall and other evaluation measures.

While there are exceptions, such as Steinberger et al., contemporary approaches to named-entity recognition are predominantly based on supervised machine learning methods. As with much work on statistical methods in natural language processing, this line of work took off in the latter half of the 1990s. Because of the sequential nature of named-entity recognition, when restricted to non-overlapping chunks, the methods proposed for the task largely follow the same trends as those for part-of-speech tagging. For example, the system of Aberdeen et al. (1995) was based on transformation-based learning (Brill, 1995), while Bennett et al. (1997) employed decision trees (Quinlan, 1986), and Bikel et al. (1999) used HMMs. As with part-of-speech tagging, these early models have now largely been surpassed by models based on CRFs, as first proposed by McCallum and Li (2003) for this task.

2.5 Sentiment

The analysis and processing of *subjective language*, as manifested for example in sentiments, beliefs and judgements, is a growing area within natural language processing. Although some interest in this area can be traced to the 1960s (Stone et al., 1966), there has been a surge of interest in the field in the last ten years, primarily thanks to the increasing significance of informal information sources, such as blogs and micro-blogs, user review sites and the booming growth of online social networks; see Pang and Lee (2008) for a comprehensive overview of this development.

Current research suggest that analyzing subjectivity in language is more difficult, compared to more traditional tasks related to content or topicality (Lee, 2004). Whether this is due to the immature nature of the field or an inherent aspect of the problem has not been settled. However, as we discuss in chapter 6, the inter-annotator agreement is typically quite low for subjective aspects of language, compared to topical aspects, which suggests that subjective language analysis is indeed an intrinsically harder problem.

Characterization

Wiebe et al. (2004), based on Quirk et al. (1985), define the term *subjective language* as referring to aspects of language use related to the expression of *private states*, such as sentiments, evaluations, emotions or speculations. A private state is characterized by a *sentiment*, possibly having a *polarity* of a certain *degree*, a *holder* and a *topic*.[8] Let the simple sentence *John likes apples a lot!* serve as an example. Here a sentiment, *liking*, is held by a holder,

[8]There is little agreement on terminology in the literature on subjective language. Common terms used for what we here call *sentiment*, include *opinion*, *attitude* and *affect*. The term *valence* is often used for what I have termed *polarity*.

John, towards a topic, *apples*. *Liking* further has a positive polarity, with a degree indicated by *a lot!*.[9] The aim of subjective language analysis is to automatically uncover aspects such as these in free text.

Most research on subjective language has focused on sentiment in isolation, or on sentiment in combination with polarity. Interest has commonly been limited to the identification of sentiment, without any further distinction between different types of sentiments, their topics or holders; and to classification of polarity into the categories of *positive*, *negative* and *neutral* (Mulder et al., 2004; Bai et al., 2005; Sahlgren et al., 2007). Thus, even when ignoring the directional aspects of holder and topic, most work has been rather coarse-grained in the characterization of private states. Some notable exceptions are the work by Bethard et al. (2004), Choi et al. (2005), Kim and Hovy (2006a), Kim and Hovy (2006b), and more recently, Johansson and Moschitti (2013), who also study methods for holder and topic identification.

Though most approaches to sentiment and polarity analysis have been coarse-grained, there has been some attempts at more fine-grained analysis. Liu et al. (2003) for example characterize sentiment in terms of Ekman's six fundamental categories of emotion: *happy, sad, angry, fearful, disgusted and surprised* (Ekman, 1993), while Subasic and Huettner (2001) use what they call "affect sets", which comprise a set of attitudinal categories with attached centralities and intensities. Other characterizations are presented in the work of Kim and Hovy (2006b), in which sentiments are characterized as *judgements* and *beliefs*, and in Thomas et al. (2006) and Kwon et al. (2007), wherein *claims* are identified and classified as to whether they are *supporting* or *opposing* an idea, or whether they are *proposing a new idea*, in the context of political discussions.

Most work in the area disregards the difficult aspect of topic and holder and instead simply analyze a piece of text as being dominated by *positive*, *negative* or *neutral* sentiment. This is the most well-studied scenario in the subjective language analysis literature and is often referred to as sentiment analysis or opinion mining (Pang and Lee, 2008). Typically, this analysis is carried out at the *word-level*, at the *document-level*, or at the *sentence-level*.

The idea of words carrying attitudinal loading is usually attributed to Osgood et al.'s theory of semantic differentiation (Osgood et al., 1967). According to this theory, meaning is defined in a multidimensional semantic space, in which dimensions are defined through pairs of antonymous adjectives and direction and distance correspond to polarity and degree, respectively. Similar ideas were embodied in the General Inquirer (Stone et al., 1966), an early system for linguistic analysis based on various lexicons. However, this is a rather crude level of analysis, since context most certainly plays a part in conveying sentiments. Still word-level sentiment analysis remains a popular approach which has been amply studied (Subasic and Huettner, 2001; Turney

[9]Note how the exclamation mark carries important information in this case.

$[\text{Rating: } 3/5]^{d}_{\text{NEU}}$

$[\text{This is my third Bluetooth device in as many years.}]^{s1}_{\text{NEU}}$
$[\text{The portable charger/case feature is great!}]^{s2}_{\text{POS}}$
$[\text{Makes the headset easy to carry along with cellphone.}]^{s3}_{\text{POS}}$
$[\text{Though the headset isn't very comfortable for longer calls.}]^{s4}_{\text{NEG}}$
$[\text{My ear starts to hurt if it's in for more than a few minutes.}]^{s5}_{\text{NEG}}$

Figure 2.5. A short product review annotated with both sentence-level and document-level sentiment. The overall product rating (3/5) is mapped to a neutral document-level sentiment.

and Littman, 2003; Riloff et al., 2003; Wiebe et al., 2004; Esuli and Sebastiani, 2006b).

Most work on subjective language analysis has been framed at the document level. Some notable examples are Pang et al. (2002), Turney (2002) and Bai et al. (2005), in which polarity classification is applied to movie reviews according to a *thumbs up*/*thumbs down* classification scheme. Pang et al. (2002) and Lee (2004) further suggest using polarity classification in business intelligence applications, by analyzing free-form survey responses and for use in recommendation systems. Other examples are Pang and Lee (2005) and Goldberg and Zhu (2006) who also classify movie reviews, but use a multi-point rating scale instead of a bipolar classification. Dave et al. (2003) perform classification of online product reviews, in addition to mining sentiments towards specific product features, while Yih et al. (2004) look at finding "hot deals" in online deal forums.

There has also been work on sentence-level analysis, for example, applied to product and movie review mining and summarization (Dave et al., 2003; Pang and Lee, 2004; Hu and Liu, 2004a,b; Popescu and Etzioni, 2005; Kim and Hovy, 2006b), classification of claims made in political discourse (Kwon et al., 2007), classification of polarity of news headlines (Sahlgren et al., 2007), and identification and analysis of judgements and beliefs (Kim and Hovy, 2006b).

In this dissertation, we will study the restricted case of product-review sentiment, where we assume that the document is assigned an overall sentiment, or *rating*, from the set {POS, NEG, NEU} (positive, negative and neutral) and each sentence of the document is assigned a sentiment, again in the set {POS, NEG, NEU}. Figure 2.5 shows an example review with both of these levels annotated. Notice how the document-level sentiment corresponds to the average of the sentence-level sentiment. In chapter 6, we exploit the correlation between the two levels, when we learn how to make predictions at the sentence-level, using the document-level rating as a training signal from which the sentence-level sentiment is induced. This is perhaps the simplest form of fine-grained sentiment analysis and one could imagine performing

a similar analysis at the clause or phrase level, as well as analyzing multiple attributes of opinions beyond their polarity (Wiebe et al., 2005). Though our methods could conceivably be applied to finer levels of granularity, for reasons of simplicity, we focus exclusively on document-level and sentence-level analysis.

Computational approaches

Most research on sentiment analysis can be categorized into one of two categories: *lexicon-centric* or *machine-learning* centric. In the former, large lists of phrases are constructed, manually or automatically, which indicate the polarity of each phrase in the list. This is typically done by exploiting common patterns in language (Hatzivassiloglou and McKeown, 1997; Riloff and Wiebe, 2003; Kaji and Kitsuregawa, 2007), lexical resources such as WordNet or thesauri (Dave et al., 2003; Hu and Liu, 2004a; Wiebe et al., 2004; Kim and Hovy, 2004; Mulder et al., 2004; Blair-Goldensohn et al., 2008; Rao and Ravichandran, 2009; Mohammad et al., 2009), or via distributional similarity (Wiebe, 2000; Turney, 2002; Sahlgren et al., 2007; Velikovich et al., 2010). In the machine-learning centric approach, one instead builds statistical text classification models based on labeled data, often obtained via consumer reviews that have been tagged with an associated star-rating (Pang et al., 2002; Pang and Lee, 2004; Gamon et al., 2005; Goldberg and Zhu, 2006; Mao and Lebanon, 2006; Blitzer et al., 2007; Snyder and Barzilay, 2007).

Both approaches have their strengths and weaknesses. Systems that rely on lexica can analyze text at all levels, including the clausal and phrasal level, which is fundamental to building user-facing technologies such as faceted opinion search and summarization (Beineke et al., 2003; Hu and Liu, 2004a; Gamon et al., 2005; Popescu and Etzioni, 2005; Carenini et al., 2006; Blair-Goldensohn et al., 2008; Titov and McDonald, 2008; Zhuang et al., 2006). However, lexica are typically deployed independent of the context in which mentions occur, which makes them brittle in the face of domain shifts and complex syntactic constructions (Wilson et al., 2005; Choi and Cardie, 2009). The machine-learning approach, on the other hand, can be trained on the millions of labeled consumer reviews that exist on review aggregation websites, often covering multiple domains of interest (Pang et al., 2002; Pang and Lee, 2004; Blitzer et al., 2007). The downside is that the supervised learning signal is often at a coarse level, most commonly the document level.

Attempts have been made to bridge this gap. The most common approach is to obtain a labeled corpus at the granularity of interest in order to train classifiers that take into account the analysis returned by a lexicon and its context (Wilson et al., 2005; Blair-Goldensohn et al., 2008). This approach combines the best of both worlds: knowledge from broad-coverage lexical resources in concert with highly tuned machine-learning classifiers that take into account context. The primary downside of such models is that they are often trained on small data sets, since fine-grained sentiment annotations

rarely exist naturally and instead require significant annotation effort per domain (Wiebe et al., 2005).

3. Structured Prediction

This chapter provides an introduction to a general framework for *structured prediction*, on which most of the methods presented in later chapters are based. In this chapter, we describe how output structures such as the linguistic structures introduced in chapter 2 can be represented and scored, in a way that allows for the optimal structure to be efficiently computed for a given input, given some learned *model parameters*. Methods for learning these model parameters from fully annotated data are described in chapter 4.

3.1 Predicting Structure

While a natural language processing system often consists of several intricate software components, we will abstract away from this complexity and instead view the system from a more formal angle. This allows us to describe different linguistic structures and systems for predicting these structures in a general and compact manner. At the most abstract level, we view the system as providing a mapping $\hat{y}: \mathcal{X} \rightarrow \mathcal{Y}$ from a set of *inputs* $\mathcal{X}$ to a set of *outputs* $\mathcal{Y}$, such that $\hat{y}(x) \in \mathcal{Y}$ is the output assigned to the input $x \in \mathcal{X}$. We assume that both the inputs and the outputs are objects with some non-trivial structure. Our goal is to induce this mapping from data, such that the predictions provided by $\hat{y}$ are as accurate as possible, according to some evaluation measure. This problem setup is referred to as *structured prediction* (Taskar, 2004; Daumé III, 2006; Bakır et al., 2007; Smith, 2011).

3.1.1 Characteristics of Structured Prediction

Many tasks traditionally addressed with machine learning methods are cast as binary, or multi-class, classification problems, where the goal is to learn to predict the correct label from a restricted set, for example, SPAM vs. NOT-SPAM in e-mail spam detection, or POLITICS vs. CULTURE vs. ECONOMY in topical text classification (Mitchell, 1997). In structured prediction, on the other hand, the output to be predicted is endowed with some internal structure and typically there is a vast number of potential outputs for any given input. Specifically, for the linguistic structures considered in this dissertation, the output may be a sequence of part-of-speech tags, a syntactic dependency tree, a segmentation into named entities or a joint assignment of sentiment to each sentence in

a product review and to the review itself. Some of these structured prediction problems can in principle be reduced to a collection of multi-class problems. For example, we could treat the problem of part-of-speech tagging as one of predicting the part of speech of each word in isolation. However, such a trivial reduction is not possible in the case of syntactic dependency parsing, where there are *structural constraints* that every dependency tree needs to obey; for example, each tree needs to be an arborescence and further constraints may apply, such as projectivity, as discussed in section 2.3.[1] Although not considered in this dissertation, it should be pointed out that structured prediction problems can also be reduced to a search process, where the prediction of search actions are cast as binary, or multi-class, classification problems (Yamada and Matsumoto, 2003; Daumé and Marcu, 2005; Daumé III et al., 2009; Ratliff, 2009).[2] The incremental structured perceptron of Collins and Roark (2004), which we employ in some of our experiments with transition-based dependency parsing, can also be viewed in this context.

Even when a naïve reduction to binary, or multi-class, classification is possible, there are reasons to treat the problem as structured, since there may be interactions between parts of the output structure that we can exploit. Consider a naturally occurring sequence of parts of speech. In such a sequence, the part of speech of a word is often highly predictive of the part of speech of the next word and certain subsequences of parts of speech are highly unlikely (Dietterich, 2002). Similarly, if a word is part of a named entity, the probability that the next word is also part of a named entity is very different from the corresponding probability when the current word is not a name. Regularities in the output structure such as these can help us make better predictions. In particular, by making structured decisions we can design our models so that observed parts of the structure can influence and constrain unobserved parts of the structure. We will use this as a key tool when learning to predict linguistic structure from partial information.

On the other hand, performing *inference* — making decisions jointly over a complex structure — is substantially more complex in structured prediction, compared to making individual binary or multi-class decisions. We will discuss inference algorithms for structured problems in more detail in section 3.5. Briefly put: although the size of the search space is typically exponential in the input size, as long as model dependencies between parts in the structure have short range, efficient inference algorithms can be derived. When taking long-range dependencies into account, we are on the other hand forced to revert to slow and/or approximate algorithms. Much current work

[1] Of course, these are constraints that have been placed by us on the *representation* of the linguistic structure at hand.

[2] Daumé III (2006) provides a more formal definition of what separates structured from non-structured prediction problems. In particular, he argues that the class of structured prediction problems should be restricted to those where the cost-function being optimized (see chapter 4) do not decompose over the output structure.

in natural language processing is devoted to devising fast and accurate approximation algorithms for complex structured prediction problems, in particular for syntactic dependency parsing; see section 2.3.

In order to make the above concepts more concrete, consider a system for part-of-speech tagging. In essence such a system takes a sentence $x \in \mathcal{X}$ as its input and returns a tagging $y \in \mathcal{Y}$ as its output, such that each of the words in the input sentence are assigned some part of speech in the output tagging. In this example, $\mathcal{X}$ is the set of all possible sentences, $\mathcal{Y}$ is the set of all possible part-of-speech tag sequences and $\hat{y}(x) \in \mathcal{Y}$ is the sequence of tags assigned to the input $x \in \mathcal{X}$. A natural measure of accuracy in this scenario is the number of correctly predicted tags in $\hat{y}(x)$, that is, the reciprocal of the Hamming distance metric; see section 4.1.2.

3.1.2 Scoring and Inference

At this stage, we place no restrictions on the input and output sets — we assume that the mapping $\hat{y}(x) \in \mathcal{Y}$ is defined for any input $x \in \mathcal{X}$ — and they may therefore both be infinite sets. Since the number of potential inputs is unbounded, it is not feasible to represent the mapping $\hat{y}$ explicitly. Rather, for each input x the system needs to perform a search over the output set $\mathcal{Y}$ to infer the most fitting output $\hat{y}(x)$. In order to facilitate this inference, we introduce a score function $s\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, such that $s(x, y)$ measures the affinity between the input x and the output y. For a fixed input x, the score function imposes a partial order on the output set, such that the relation $s(x, y) > s(x, y')$ corresponds to the belief that the output y is more accurate than output y' for the input x, while $s(x, y) = s(x, y')$ indicates no preference between y and y'. For a given input x, we thus predict the output $\hat{y}(x)$ with the highest score:

$$\hat{y}(x) = \arg\max_{y \in \mathcal{Y}} s(x, y)\,. \tag{3.1}$$

In this chapter, we will consider the score function as fixed and given and we will focus on a representation of inputs and outputs that facilitates an efficient solution of the inference problem in eq. (3.1), while the remainder of the dissertation is focused on different scenarios for *learning* the score function from data, such that the mapping $\hat{y}$ is as accurate as possible, according to some problem specific evaluation measure.

Clearly, although the score function provides a partial order on the output set for a given input, it is not feasible to explore the infinite output set. However, since the input will always be given, we only need to consider those outputs that are *valid* for a particular input. Consider the part-of-speech tagging example again. In this case we know from the problem definition that the output should assign exactly one tag to each token. Consequently, we only need to consider tag sequences of the same length as the input. Let

$\mathcal{Y}(x) \subset \mathcal{Y}$ denote the set of valid outputs for an input $x \in \mathcal{X}$ and let us restrict the mapping $\hat{y}$ accordingly, so that $\hat{y}(x) \in \mathcal{Y}(x)$. We refer to the set $\mathcal{Y}(x)$ as the *inference space* and we say that the structures $y \in \mathcal{Y}(x)$ *span* the input x. By enforcing that $s(x, y) = -\infty$ for every output $y \notin \mathcal{Y}(x)$, eq. (3.1) can be reformulated as

$$\hat{y}(x) = \arg\max_{y \in \mathcal{Y}} s(x, y) = \arg\max_{y \in \mathcal{Y}(x)} s(x, y)\,. \tag{3.2}$$

While it is possible to define structures such that the inference space is also infinite, for example, by allowing unbounded recursion, in this dissertation, we will only consider structures for which the inference space is a finite set. However, its size will typically be exponential in the size of the input, which still poses a challenge for the construction of efficient inference algorithms. Depending on the structure of the inference space, computing the arg max in eq. (3.2) can be more or less computationally demanding. For all of the models studied in this dissertation there are well-known efficient dynamic programming algorithms with polynomial time and space requirements. We will briefly discuss different approaches to inference in section 3.5.

3.2 Factorization

Thus far, we have viewed the input and output spaces as abstract sets, with elements of the output set being (implicitly) partially ordered by the score function. From this perspective, there is no relationship between elements in the output set, except for their order with respect to a specific input. In order to facilitate the construction of efficient algorithms for exploring the typically exponential inference space, we need to impose some additional structure on the output space. To this end, we will henceforth assume that each structure $y \in \mathcal{Y}$ can be decomposed into a set of smaller and simpler, potentially overlapping, substructures. We already touched upon this idea in the part-of-speech tagging example in the previous section, where we described a sequence of tags spanning the input as being decomposable into a set of position-specific tags. This decomposition of the tagging problem is, however, not the only possible. For example, we could also decompose the sequence of tags into sets of partially overlapping position-specific tag n-grams, that is, into partially overlapping subsequences of n consecutive tags. The latter decomposition has the advantage over the former that it allows for modeling interactions between consecutive tags, such as the fact that adjectives have a tendency to be followed by nouns in English. At the same time, by allowing for partially overlapping substructures, the search for an optimal sequence of tags is complicated, due to the constraint that the overlapping tag n-gram assignments need to be consistent across the whole tag sequence. The manner in which we decompose the outputs is of crucial importance for the development of efficient inference algorithms, as we will discuss later.

A natural way to describe the class of decomposable discrete structures just sketched is by means of binary indicator vectors indexed by means of some structured index set (Smith, 2011). Let $\mathcal{I}$ be an index set whose structure is determined by the particular decomposition of the outputs. We define the output set $\mathcal{Y}$ as the set of structures that can be assembled from substructures indexed by $\mathcal{I}$. Correspondingly, $\mathcal{Y}(x)$ is defined in terms of $\mathcal{I}(x)$, where $\mathcal{I}(x)$ is restricted to indices of substructures in the span of x. The set of all indicator vectors indexed by indices from the index set $\mathcal{I}$ is $\{0,1\}^{|\mathcal{I}|}$. For a given output $y \in \mathcal{Y}$, we use the convention that $y(i) = 1$ for any index $i \in \mathcal{I}$ if and only if the ith substructure is a part of the output y and we will use the notation $i \in y$ to refer to those indices that correspond to active substructures of y, that is, $\{\, i : i \in y \,\} = \{\, i : i \in \mathcal{I} \wedge y(i) = 1 \,\}$. However, since there are typically some constraints on the elements of $\mathcal{Y}$, for example, arborescence constraints in the case of dependency trees, not every binary index vector corresponds to a valid output and consequently, $\mathcal{Y}(x) \subset \{0,1\}^{|\mathcal{I}(x)|}$.

At this point it may not be clear what this decomposition into substructures affords us. We have mentioned that this allows us to search the inference space more efficiently. However, this is only true for certain benign combinations of index sets and output constraints. In the most general case, finding the highest scoring output is an NP-hard problem. In addition to a benign combination of index set and constraints, in order to achieve efficient inference, we need the score function to decompose into scores of substructures, according to such an index set. Before discussing these issues, we discuss how the linguistic structures from chapter 2 can be represented by means of index sets. Following this, in section 3.3, we discuss how to express the score function in terms of salient *features* of the input and the output.

3.2.1 Sequence Labeling

Part-of-speech tagging and named-entity recognition (described in section 2.2 and in section 2.4, respectively), can both be formulated as *sequence-labeling* problems (Ratnaparkhi, 1996). We assume that the input text has been split into sentences and that each sentence has been tokenized. Our task is to assign a tag from the provided tag set to each lexical item (word/token) of a given text. While possible (Rush et al., 2012), we do not assume that there are any dependencies between sentences, so that we can process each sentence separately. We cast this problem in the general structured prediction framework as follows. Let $x = (x_1 x_2 \ldots x_{|x|}) \in \mathcal{X}$ denote a sentence, where x_i denotes the ith token in x. Let $y = (y_1 y_2, \ldots y_{|x|}) \in \mathcal{Y}(x)$ denote a tag sequence, where $y_i \in \mathcal{T}$ is the tag assigned to the ith token in x and $\mathcal{T}$ is the tag set used. The space of tag sequences $\mathcal{Y}(x)$ can be expressed in terms of the index set

$$\mathcal{I}_{\text{seq}}(x) \;= \{\, (i,t) : i \in [1,|x|],\; t \in \mathcal{T} \,\},$$

where $y(i, t) = 1$ has the interpretation that the ith token in the sequence is assigned the tth tag, while $y(i, t) = 0$ means that the ith token is not assigned the tth tag. We further have the constraint that each position is to be assigned exactly one tag. In practice, we make sure that this and other structural constraints, such as arborescence and projectivity constraints in syntactic dependency parsing, are obeyed by structuring the inference algorithm such that only outputs that obey the constraints are explored; see section 3.5.

For now, we will leave details of language specific part-of-speech tag sets aside and simply assume that we are provided with some set of tags $\mathcal{T}$. As an example, the coarse-grained "universal" part-of-speech tag set of Petrov et al. (2012) consists of the following twelve tags: NOUN (nouns), VERB (verbs), ADJ (adjectives), ADV (adverbs), PRON (pronouns), DET (determiners and particles), ADP (prepositions and postpositions), NUM (numerals), CONJ (conjunctions), PRT (particles), PUNC (punctuation) and X (a catch-all category for abbreviations, foreign words and so on). In named-entity recognition with the CoNLL tag set (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) and a BIO encoding, we have the tag set $\mathcal{T} = \{\text{O}\} \cup \{\text{B-X}, \text{I-X}\}$ for $\text{X} \in \{\text{PER}, \text{ORG}, \text{LOC}, \text{MISC}\}$ (outside and begin/inside person, organization, location, and miscellaneous).

The index set $\mathcal{I}_{\text{seq}}(x)$ corresponds to a zeroth-order *Markov factorization*, which is rich enough to express the space of all possible tag-sequences. When the score function is factorized in terms of this index set, we have a model in which each tag is independent of all other tags. However, in natural languages, there are often strong correlations between substructures. For example, in English part-of-speech sequences, determiners tend to be immediately followed by nouns or adjectives, while adjectives tend to be followed by nouns, and so on. In order to exploit such regularities of naturally occurring sequences, we can instead make a kth-order Markov assumption when factorizing the score function. This corresponds to the index set

$$\mathcal{I}^k_{\text{seq}}(x) \;= \{\, (i, t_{i-k:i}) : i \in [1, |x|],\; t_{i-k:i} \in \mathcal{T}^{k+1} \,\},$$

where the structured index $(i, t_{i-k:i})$ represents a sequence of $k + 1$ tags assigned to $y_{i-k:i}$. In terms of indicator vectors, we thus have that $y_{i-k:i} = t_{i-k:i} \Leftrightarrow y(i, t_{i-k:i}) = 1$.

Performing inference with the higher-order index set is more computationally demanding, because we need to take into account the fact that the score function is expressed in terms of overlapping tag subsequences. Note that $\mathcal{I}_{\text{seq}} = \mathcal{I}^0_{\text{seq}}$ and that each index in a lower-order factorization subsumes a set of indices in a higher-order factorization. This hierarchical relationship between the index sets of various orders is exploited by *coarse-to-fine* methods (Charniak and Johnson, 2005; Weiss and Taskar, 2010; Rush and Petrov, 2012), where efficient lower-order models are used to prune the search space of higher-order models.

Joint document- and sentence-level sentiment analysis (see section 2.5) can also be cast as a variant of sequence labeling. In this case, rather than having sentences represented as sequences of tokens, we represent a document as a sequence of sentences. Additionally, we model the document-level by adding an extra sub-index that represents the sentiment label assigned at this level. Thus, let $x = (x_1, x_2, \ldots, x_{|x|})$ be a document composed of a sequence of sentences x_i and let $y = (y^d, y^s_1 y^s_2 \ldots y^s_{|x|}) \in \mathcal{Y}(x)$, where y^d denotes the document-level sentiment variable and y^s_i denotes the ith sentence-sentiment variable. The inference space is defined by the index set

$$\mathcal{I}_{\text{sent}}(x) = \{\, (i, s, d) : i \in [1, |x|],\ s \in \mathcal{S}_s,\ d \in \mathcal{S}_d \,\},$$

where i ranges over the sentence positions in the document, s ranges over the set of sentence-level sentiment labels $\mathcal{S}_s$ and d ranges over the set of document-level sentiment labels $\mathcal{S}_d$. While other label sets are possible, we will use the same set of labels for both the document level and the sentence level. Specifically, we let $\mathcal{S} = \mathcal{S}_s = \mathcal{S}_d = \{\text{POS}, \text{NEG}, \text{NEU}\}$ (positive, negative and neutral).

Note that the document-level sentiment is fixed for each sequence of assignments to the sentence-level. Thus, for each fixed value of the document-level sentiment, the problem is reduced to a standard sequence-labeling problem. Just as with part-of-speech tagging and named-entity recognition, there may be sequential regularities in the data. Based on intuitions about discourse structure, we assume that sentences expressing a certain sentiment are clustered, so that positive sentences and negative sentences tend to cluster in different parts of the document. In order to exploit these posited regularities, we can again use a kth-order Markov factorization, corresponding to the index set

$$\mathcal{I}^k_{\text{sent}}(x) = \{\, (i, s_{i-k:i}, d) : i \in [1, |x|],\ s_{i-k:i} \in \mathcal{S}_s^{k+1},\ d \in \mathcal{S}_d \,\}.$$

3.2.2 Arc-Factored Dependency Parsing

We now turn to the framing of graph-based syntactic dependency parsing (see section 2.3) as a structured prediction problem. Let $x = (x_1 x_2 \ldots x_{|x|})$ denote an input sentence, where x_i denotes the ith token and let $y \in \mathcal{Y}(x)$ denote a dependency tree, where $\mathcal{Y}(x)$ is the set of well-formed dependency trees spanning x.

As discussed in section 2.3, we will only consider first-order arc-factored models (Eisner, 1996; McDonald et al., 2005) in this dissertation. In the case of *labeled* dependency parsing, where each head-dependent relation is labeled with its grammatical relation, we can represent the inference space of the first-order factorization by means of the labeled arc index set

$$\mathcal{I}_{\text{arc}}(x) = \{\, (h, d, r) : h \in [0, |x|],\ d \in [1, |x|],\ r \in \mathcal{R} \,\},$$

where the head-index h ranges over the token positions (including the zeroth position, which correspond to the dummy ROOT token), the dependent-index d ranges over the token positions (excluding the ROOT token) and r ranges over the set of grammatical relations $\mathcal{R}$.

Different syntactic annotations define different relations. For example, the annotation guidelines for the *Stanford typed dependencies* (de Marneffe and Manning, 2008) defines 53 different grammatical relations. For practical reasons, discussed in more detail in chapter 7, we focus on unlabeled dependencies in the multilingual scenario, because the syntactic annotations in different languages use different sets of grammatical relations, which makes it difficult to evaluate and compare labeled models. For the corresponding *unlabeled* arc-factored model, the relation label sub-index is simply dropped in the unlabeled arc index set

$$\mathcal{I}_{\text{uarc}}(x) = \{ (h, d) : h \in [0, |x|],\ d \in [1, |x|] \}$$

The first-order factorization allows us to express all possible dependency trees. However, it assumes that the score of a tree factors into scores of individual arcs. This factorization is not powerful enough to reach state-of-the-art results in the fully supervised case, because it cannot exploit any substructure regularities, such as those between sibling arcs and between parent and grandparent arcs (McDonald and Pereira, 2006; Carreras, 2007; Smith and Eisner, 2008; Koo and Collins, 2010; Zhang and McDonald, 2012). However, in the scenarios that we consider in this dissertation, the performance is not yet at the level of supervised parsing and the first-order factorization is therefore sufficient for our purposes.

3.3 Parameterization

With the representation of inputs and outputs in place, let us discuss the score function $s(x, y)$ in more detail. In order to achieve efficient inference and learning, we seek a score function with the following three properties.

First, the score function should be amenable to efficient inference. That is, we seek a score function that allows us to explore the inference space as efficiently as possible to find, for example, the output with the highest score, or the output with the k highest scores, for a particular input. One step towards achieving this is by assuring that the score function decomposes over the index set, that is, the score of an input-output pair can be expressed in terms of the scores of substructures — additionally, the combination of index set and output constraints should be such that the assembly of the scored substructures into a coherent output can be performed efficiently.

Second, the score function should be amenable to learning in the sense that it can be adapted to the data. For learning to be tractable, the mapping from inputs to outputs should behave nicely, in the sense that changes in

the score function leads to predictable changes in the mapping. As discussed in chapter 4, mathematically, this equates to the requirement that the score function — in combination with a task specific cost function — should at least be sub-differentiable with respect to its parameters.

Third, the score function should be able to compactly represent the affinities of an infinite number of input-output pairs. As briefly discussed in chapter 4, such a compact representation is not only preferable for reasons of computational efficiency and storage efficiency; from a learning theoretic perspective, a compact representation that still fits the data well tends to lead to better generalization ability (Vapnik, 1998).

The simplest score function that satisfies these properties is a function linear in the representation of input-output pairs, such that the representation decomposes over the index set.[3] In order to define such a score function, our first step is to represent pairs of inputs and outputs by means of their salient features. This is accomplished with a vector-valued *feature function* $\Phi: \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$, which maps each input-output pair to a d-dimensional real-valued vector.[4] We then parameterize the score function by a *parameter vector* $\theta \in \Theta$. This gives us the score function $s: \mathcal{X} \times \mathcal{Y} \times \Theta \to \mathbb{R}$, where $s_\theta(x, y)$ denotes the score of $(x, y) \in \mathcal{X} \times \mathcal{Y}$, given parameters θ. Throughout, we shall restrict ourselves to the case $\Theta = \mathbb{R}^d$ and exclusively consider the linear score function

$$s_\theta(x, y) = \theta^\top \Phi(x, y) = \sum_{j=1}^{d} \theta_j \Phi_j(x, y) \,. \tag{3.3}$$

In other words, each feature $\Phi_j(x, y)$ encodes a salient feature of the input-output pair (x, y), and each feature is paired with a parameter θ_j. If we assume that the elements of the feature vector are all non-negative, for a fixed input, we can increase the scores of those outputs for which the feature is active by increasing the weight of the corresponding parameter. Analogously, we can decrease the scores of the same outputs, by decreasing the weight of the parameter. Thus, for a fixed feature representation, the score function is solely controlled by its parameters.

In order to pave the way for efficient inference, we further restrict the feature function to decompose linearly over the index set $\mathcal{I}$:

$$\Phi(x, y) = \sum_{i \in \mathcal{I}(x)} \phi(x, y, i) \cdot \mathbb{1}\left[y(i) = 1\right] = \sum_{i \in y} \phi(x, y, i) \,,$$

where $\phi(x, y, i) \in \mathbb{R}^d$ is a substructure feature function that extracts a feature vector representation of the combination of the input and the ith output

[3] The compactness property is not inherently satisfied by linear functions, but this can be achieved by adding an appropriate regularizer during learning (see section 4.1.1).

[4] In practice, we only use feature vectors with binary elements. Categorical features with K potential values are trivially binarized by using K binary elements, while real-valued features can first be appropriately "bucketed" into categorical features, which can then be binarized.

substructure and $\mathbb{1}[\cdot]$ is the indicator function that evaluates to 1 when its argument is true and to 0 otherwise. This gives the following complementary decomposition of the score function in terms of scores of substructures:

$$s_\theta(x, y) = \sum_{i \in y} \theta^\top \phi(x, y, i) = \sum_{j=1}^{d} \sum_{i \in y} \theta_j \phi_j(x, y, i) \,.$$

The specific form of the substructure feature function $\phi(x, y, i)$ is application and model specific. The structured index i is used to address the parts of the input-output pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ that are accessible to $\phi(x, y, i)$. The substructure feature function is defined locally with respect to the index i and cannot encode information that requires access to anything outside the ith part of the output. In *discriminative models*, which are the type of models used primarily in this dissertation, each substructure feature function can access the whole input, while in *generative models* each substructure feature function is restricted to be local both in the input and the output; see sections 3.4.1 and 3.4.2. In the latter case, the structured index i is thus used to address both parts of the input and parts of the output. Often, we can further factor the substructure feature function into an input part $\varphi(x, i)$ and an output part $\psi(y, i)$:

$$\phi(x, y, i) = \varphi(x, i) \otimes \psi(y, i) \,,$$

where $a \otimes b$ denotes the outer-product of the vectors $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$, followed by a projection that maps the outer product matrix to an $m \times n$-dimensional vector by concatenating the row vectors of the matrix. The upshot of this factorization, in terms of computational efficiency, is discussed further in section 3.5.

3.4 Probabilistic Models

The score function in eq. (3.3) maps each input-output pair to a real-valued scalar. The scores are thus finite, but can grow without bounds in both the positive and the negative direction. Moreover, the scores assigned to outputs given a particular input x are not directly comparable with the scores assigned to outputs given another input x'. This makes it difficult to use these scores for things like confidence estimation. It also makes the scores less adequate for use in *ensemble methods*, Dietterich (2000), which combine the predictions of different models for the same input; a technique that has been shown to often improve results for various prediction tasks.

Probabilistic models provide a natural framework for reasoning about notions such as confidence and for performing model combination. Technical details aside, a discrete probability distribution $p(E)$ with respect to a random

variable E taking values $e \in \mathcal{E}$ is a real-valued function with the property that

$$p(E = e) \geq 0\,, \forall e \in \mathcal{E} \qquad \text{and} \qquad \sum_{e \in \mathcal{E}} p(E = e) = 1\,.$$

For ease of notation, we will subsequently take the random variable as implicit and simply write $p(e)$ in place of $p(E = e)$. Treating the input and output as random variables X and Y, respectively, let us rewrite the non-probabilistic score function in terms of these variables as $s(X, Y)$. For an arbitrary combination of input and output, that is, for any assignment $X = x$ and $Y = y$, the score $s(x, y)$ may grow unbounded. In contrast, the probability distribution $p(X, Y)$ obeys the constraint that $p(x, y) \in [0, 1]$ for all $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Additionally, the distribution is normalized so that $\sum_{x \in \mathcal{X}, y \in \mathcal{Y}(x)} p(x, y) = 1$.

The distribution $p(X, Y)$ is known as a *joint* distribution, because it models both the input and the output as random variables. Such distributions are fundamental to *generative* models, which are commonly employed in *unsupervised* and *semi-supervised* learning. In this dissertation, our focus will mostly be on models based on a *conditional* distribution $p(Y \mid X)$, where only the output is treated as a random variable, while the input is treated as fixed and always observed. Models based on conditional distributions are employed in *discriminative* models. For further discussion on the generative-discriminative distinction, see Ng and Jordan (2001); Minka (2005); Lasserre et al. (2006).

The score function $s(x, y)$ makes no distinction between joint and conditional models; it is simply defined as a function with domain $\mathcal{X} \times \mathcal{Y}$. The reason that we need to make this distinction explicit in probabilistic models is the constraint that the distribution has to sum to one over its domain, that is, for a joint model it is required that

$$\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}(x)} p(x, y) = 1\,,$$

while in a conditional model, we have the constraint that

$$\sum_{y \in \mathcal{Y}(x)} p(y \mid x) = 1\,, \forall x \in \mathcal{X}\,.$$

We will return to these concepts in chapter 5, when we discuss learning from partial information.

3.4.1 Globally Normalized Models

We focus on discrete distributions that can be expressed in terms of linear score functions. However, rather than letting the parameterized distribution $p_\theta(Y \mid X)$ be a function linear in its parameters θ, we let $\log p_\theta(Y \mid X)$ be

linear in θ. In the conditional case, the resulting distribution has the form

$$p_\theta(y \mid x) = \frac{\exp\{s_\theta(x, y)\}}{Z_\theta(x)}, \tag{3.4}$$

where the so called *partition function* $Z_\theta(x)$ is defined as

$$Z_\theta(x) = \sum_{y' \in \mathcal{Y}(x)} \exp\{s_\theta(x, y')\}. \tag{3.5}$$

In the joint case, the distribution has the same form, with the difference that the partition function sums over both $\mathcal{X}$ and $\mathcal{Y}(x)$:

$$p_\theta(x, y) = \frac{\exp\{s_\theta(x, y)\}}{Z_\theta}, \tag{3.6}$$

where

$$Z_\theta = \sum_{x' \in \mathcal{X}} \sum_{y' \in \mathcal{Y}(x)} \exp\{s_\theta(x', y')\}. \tag{3.7}$$

The exponentiation of the score function makes sure that the distribution is non-negative, while normalizing by the partition function ensures that the probability mass sums to one over the support of the distribution. The family of distributions of the form in eq. (3.4) is known as the *exponential family* in the statistics community (Wainwright and Jordan, 2008), while the terms *log-linear* model and *maximum-entropy* model are more common in the natural language processing community (Berger et al., 1996; Ratnaparkhi, 1996).

A nice aspect of these models is that we can use the underlying score function directly for prediction. Assume that we want to predict the most probable output $\hat{y}(x) \in \mathcal{Y}(x)$ for some input $x \in \mathcal{X}$. Consider a conditional model, where the probability of an arbitrary output $y \in \mathcal{Y}(x)$ is given by

$$p_\theta(y \mid x) \propto \log p_\theta(y \mid x) = s_\theta(x, y) - \log Z_\theta(x) \propto s_\theta(x, y).$$

The most probable output is thus strictly a function of the score $s_\theta(x, y)$:

$$\hat{y}(x) = \arg\max_{y \in \mathcal{Y}(x)} p_\theta(y \mid x) = \arg\max_{y \in \mathcal{Y}(x)} s_\theta(x, y).$$

Hence, for a fixed parameter vector θ, the probabilistic model predicts the same output as its non-probabilistic counterpart. Since computing the partition function is typically more computationally demanding, compared to only computing the highest scoring output, this is a useful correspondence.

The distributions in eqs. (3.4) and (3.6) are *globally normalized* distributions, in the sense that the partition function sums over complete outputs. When the output factorization has a graph structure, these correspond to *undirected graphical models* (Koller and Friedman, 2009). Specifically, in this case the conditional model in eq. (3.4) defines a *conditional random field* (CRF;

Lafferty et al., 2001), while eq. (3.6) defines a *Markov random field* (MRF; Kindermann and Snell, 1980).[5]

3.4.2 Locally Normalized Models

In *locally normalized* models, the (joint or conditional) distribution factorizes into a product of local probability distributions; that is, the distribution at each local factor is normalized. Typically, when a sufficient amount of labeled training data is available, globally normalized models are more powerful, because they make less unwarranted independence assumptions. In particular, globally normalized conditional models tend to outperform locally normalized joint models, primarily because the former can use richer feature definitions, such as features defined with respect to the whole input and parts of the output jointly. In this dissertation, we will only consider locally normalized distributions in the context of part-of-speech tagging. Specifically, we employ hidden Markov models (HMMs; Rabiner, 1989) for this problem in chapter 10, where we show that globally normalized models can be more effective than locally normalized models, even when only partially annotated data is available for training.[6]

Let $x = (x_1 x_2 \dots x_{|x|}) \in \mathcal{X}$ denote a sequence of observations, where each observation $x_i \in \mathcal{V}$ is an instance of a type from the *vocabulary* $\mathcal{V}$. Further, let $y = (y_1 y_2 \dots y_{|x|}) \in \mathcal{Y}$ denote a tag sequence, where $y_i \in \mathcal{T}$ is the tag assigned to observation x_i and $\mathcal{T}$ denotes the set of all possible tags. When used for part-of-speech tagging, each observation x_i is a token and the vocabulary enumerates all word types (not necessarily restricted to types observed during training). A kth-order HMM for sequence labeling corresponds to the index set

$$\mathcal{I}^k_{\text{hmm}}\{\,(i, v, t_{i-k:i}) : i \in [1, |x|],\ v \in \mathcal{V},\ t_{i-k:i} \in \mathcal{T}^{k+1}\,\},$$

with the interpretation $x(i, v) = 1 \Leftrightarrow x_i = v$ and $y(i, t_{i-k:i}) = 1 \Leftrightarrow y_{i-k:i} = t_{i-k:i}$. The joint distribution over $\mathcal{X}$ and $\mathcal{Y}$ factorizes into the product of *emission* distributions and *transition* distributions:

$$p_\beta(x, y) = \prod_{i=1}^{|x|} \underbrace{p_\beta(x_i \mid y_i)}_{\text{emission}} \underbrace{p_\beta(y_i \mid y_{i-k:i-1})}_{\text{transition}}. \tag{3.8}$$

Traditionally, HMMs have been formulated in terms of *categorical* distributions. In this case, the component distributions corresponds to conditional probability tables $\beta_{v,t} \in [0, 1]$ for all $(v, t) \in \mathcal{V} \times \mathcal{T}$ representing the emission distribution and $\beta_{t, t_{-k:-1}} \in [0, 1]$ for all $(t, t_{-k:-1}) \in \mathcal{T} \times \mathcal{T}^k$ representing

[5] This terminology is slightly unfortunate, since Markov assumptions are typically used with both CRFs and MRFs. The term *conditional Markov random field* would thus be more appropriate.

[6] Not that an HMM is both locally and globally normalized, since the local normalization property, together with the model structure, in this case guarantees global normalization.

the transition distribution, with the constraints that $\sum_{v \in \mathcal{V}} \beta_{v,t} = 1$ for all $t \in \mathcal{T}$ and $\sum_{t \in \mathcal{T}} \beta_{t,t_{-k:-1}} = 1$ for all $t_{-k:-1} \in \mathcal{T}^k$. However, recent work has shown that using log-linear component distributions, as proposed by Chen (2003), gives superior prediction performance in many natural language processing applications (Berg-Kirkpatrick et al., 2010; Das and Petrov, 2011; Li et al., 2012). This parameterization is preferable, because it allows for the model parameters β to be shared across categorical events, which in particular increases the model's ability to generalize to observations that were not seen during training.

With a log-linear parameterization, the categorical emission and transition events are instead represented by vector-valued feature functions $\phi(x_i, y_i)$ and $\phi(y_i, y_{i-1}, \ldots, y_{i-k})$. The parameters now form a vector β, in which each element corresponds to a particular feature, and the log-linear component distributions have the form

$$p_\beta(x_i \mid y_i) = \frac{\exp\left\{\beta^\top \phi(x_i, y_i)\right\}}{\sum_{x_i' \in \mathcal{V}} \exp\left\{\beta^\top \phi(x_i', y_i)\right\}}$$

and

$$p_\beta(y_i \mid y_{i-k:i-1}) = \frac{\exp\left\{\beta^\top \phi(y_i, y_{i-1}, \ldots, y_{i-k})\right\}}{\sum_{y_i' \in \mathcal{T}} \exp\left\{\beta^\top \phi(y_i', y_{i-1}, \ldots, y_{i-k})\right\}}\,.$$

3.4.3 Marginalization and Expectation

In addition to the most likely output (and its probability) for some input, we will be interested in the *expectation* of some function of a random variable with respect to a particular distribution over that variable, as well as in the *marginal probability* of a (set of) substructure(s).

Let E be a random variable taking values $e \in \mathcal{E}$. The expectation of a function $f(e)$ with respect to a discrete distribution $p(e) = p(E = e)$ is defined as

$$\mathop{\mathbb{E}}_{p(e)}\left[f(e)\right] = \sum_{e \in \mathcal{E}} f(e)p(e)\,.$$

Let $\{E_1, E_2, \ldots, E_n\}$ be a collection of random variables taking values $e_i \in \mathcal{E}_i$ for $i = 1, 2, \ldots, n$ and let their joint distribution be $p(e_1, e_2, \ldots, e_n)$. The marginal distribution of the ith variable is obtained by *marginalizing* — that is, summing — over the joint distribution over all possible assignments of the remaining variables:

$$p(E_i = e) = \sum_{\{(e_1, e_2, \ldots, e_n) : (e_1, e_2, \ldots, e_n) \in \mathcal{E}_1 \times \mathcal{E}_2 \times \ldots \times \mathcal{E}_n \wedge e_i = e\}} p(e_1, e_2, \ldots, e_n)$$

Since the output $y \in \mathcal{Y}(x)$ for an input $x \in \mathcal{X}$ corresponds to a joint assignment of the binary substructure variables indexed by $\mathcal{I}(x)$, the marginal

probability of substructure $i \in \mathcal{I}(x)$ being active is

$$p(i \mid x) = p(y(i) = 1 \mid x) = \sum_{\{\, y : y \in \mathcal{Y}(x) \wedge y(i) = 1 \,\}} p(y \mid x) = \sum_{y \in \mathcal{Y}(x)} y(i) p(y \mid x) \,.$$

This marginal probability can also be defined in terms of the expectation of the indicator function $y(i)$ with respect to the distribution $p_\theta(y \mid x)$:

$$p_\theta(i \mid x) = \mathop{\mathbb{E}}_{p_\theta(y \mid x)} \big[y(i) \big] \,.$$

When learning with probabilistic conditional models (see section 4.2.2), we will be interested in the expectation of the feature function $\Phi(x, y)$ with respect to $p_\theta(y \mid x)$:

$$\mathop{\mathbb{E}}_{p_\theta(y \mid x)} \big[\Phi(x, y) \big] = \sum_{y \in \mathcal{Y}(x)} \Phi(x, y) p_\theta(y \mid x) \,.$$

Since the feature function factors according to the index set $\mathcal{I}(x)$, we will in particular be interested in the expectation of the substructure feature function $\phi(x, y, i)$ with respect to $p_\theta(y \mid x)$:

$$\mathop{\mathbb{E}}_{p_\theta(y \mid x)} \big[\phi(x, y, i) \big] = \sum_{y \in \mathcal{Y}(x)} \phi(x, y, i) y(i) p_\theta(y \mid x) = \sum_{y \in \mathcal{Y}(x)} \phi(x, y, i) p_\theta(i \mid x) \,.$$

3.5 Inference

The methods that we consider in this dissertation require the solution of two inference problems. First, in order to perform prediction with a fixed set of parameters, we need to be able to efficiently compute the arg max in eq. (3.1). Second, when working with probabilistic models, we need to compute marginal distributions over substructures as described in the previous section. In order to compute the marginals of globally normalized models, we also need to efficiently compute the partition function; see eqs. (3.5) and (3.7).

In the most general case, with a linear score function and linear output constraints, finding the arg max corresponds to solving an integer linear program (ILP). There are many off-the-shelf ILP solvers that could potentially be used for general inference. Unfortunately, computing the solution of a general ILP is an NP-hard problem. Nevertheless, such an approach has been successfully used for a variety of natural language processing tasks (Punyakanok et al., 2005; Riedel and Clarke, 2006; Denis and Baldridge, 2007; Martins et al., 2009).

For the models that we consider in this dissertation, efficient exact algorithms are available that exploit the specific form of the inference space. For sequence-labeling models with a kth-order Markov factorization, the highest scoring tag sequence can be found with dynamic programming (Bellman,

1957) in $\mathcal{O}(l^{k+1}n)$ time, where l is the number of tags in the tag set and n is the length of the tag sequence, using the classic Viterbi algorithm (Viterbi, 1967). Similarly, with these models, the partition function and marginal probabilities can be computed with a dynamic programming algorithm known as the forward-backward algorithm (Baum, 1972). The asymptotic time complexity of the forward-backward algorithm is the same as that of the Viterbi algorithm. These algorithms can be generalized to exact algorithms for tree structured graphical models, where they are known as the max-product algorithm and the sum-product algorithm, respectively (Yedidia et al., 2003). Furthermore, these algorithms can both be expressed as special cases of dynamic programming algorithms over graphs, where the edge weights and path aggregation operations constitute a semiring (Goodman, 1999; Mohri, 2002; Eisner, 2002).

When restricted to projective dependency trees, inference in labeled arc-factored models can be solved with variants of the bottom up dynamic programming algorithm of Eisner (1996). For arc-factored models, the computational complexity is $\mathcal{O}(n^3l)$, where n is the number of tokens in the sentence and l is the number of grammatical relations. The derivations performed in Eisner's algorithm can be represented as a directed hypergraph (Gallo et al., 1993; Klein and Manning, 2001) on which inference of the arg max and the partition function can be performed with a generalization of the Viterbi algorithm, using different semirings (Huang, 2008a). Similarly, marginals can be computed efficiently with a generalization of the inside-outside algorithm (Baker, 1979) to hypergraphs.

4. Statistical Machine Learning

In chapter 3, we described how to represent different types of linguistic structure and how to compute the most likely structure for an input, given some model parameters. In this chapter, we describe how to learn these model parameters from fully annotated data. Methods for learning from partial information are discussed in chapter 5, while methods for cross-lingual learning and prediction are discussed in chapter 7. After introducing the framework of regularized empirical risk minimization, we discuss different cost/loss functions and regularizers. Following this, we discuss gradient-based optimization methods and finally we give some tricks of the trade that we use to implement the discussed algorithms efficiently.

4.1 Supervised Learning

We now describe the fully supervised learning scenario in more detail. Let us start by recapitulating the concepts and the notation from chapter 3. We denote an input by $x \in \mathcal{X}$ and an output by $y \in \mathcal{Y}$ and we assume that there is some efficient way of enumerating all valid outputs $\mathcal{Y}(x)$ spanning an input x. The goal of supervised learning is to learn a mapping $\hat{y}: \mathcal{X} \to \mathcal{Y}$ from a finite sample of input-output pairs, such that the mapping is optimal according to some cost function. While $\mathcal{X}$ and $\mathcal{Y}$ may have arbitrary complex structure, the input-output pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ are mapped into a vector space by means of a task-specific joint feature map $\Phi: \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d_+$. Moreover, we assume a vector $\theta \in \Theta$ of model parameters, with $\Theta = \mathbb{R}^d$, where parameter θ_j is paired with feature $\Phi_j(\cdot)$. Finally, the score function $s: \mathcal{X} \times \mathcal{Y} \times \Theta \to \mathbb{R}$ is taken to be the linear function $s_\theta(x, y) = \theta^T \Phi(x, y)$, throughout. Each value of the parameter vector $\theta \in \Theta$ corresponds to a mapping $\hat{y}_\theta: \mathcal{X} \times \Theta \to \mathcal{Y}$, via the function

$$\hat{y}_\theta(x) = \arg\max_{y \in \mathcal{Y}(x)} s_\theta(x, y) \,.$$

4.1.1 Regularized Empirical Risk Minimization

This representation provides us with a machinery for scoring potential outputs for a given input, such that we can increase or decrease the scores of different outputs for a given input, by changing the elements of the model parameters θ. It also endows the input-output space with a geometry, which

allows us to talk about distances between pairs of inputs and outputs, for example, via the Euclidian metric. However, for learning to be possible, we need a way to signal to the learner which outputs are preferable and which are not, so that it can adapt the parameters to better fit the data. This is accomplished by a task specific cost function $C\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$, where $c(y, y')$ measures the cost of predicting y' when the correct output is y. We assume that this cost is always finite, that $C(y, y) = 0$ and that $C(y, y') > 0$ for all $y \neq y'$. Overloading notation, we let $C(x, y, \theta) = c(y, \hat{y}_\theta(x))$ be a measure of the cost of using the parameters θ for prediction, when the input is x and the correct output is $y \in \mathcal{Y}(x)$. We next assume that there is some (unknown) distribution $p(X, Y)$ according to which input-output pairs (x, y) are distributed. We can think of this data-generating distribution as "reality", such that the probability of a pair (x, y) occurring in "the real world" is given by $p(x, y)$.

With these preliminaries in place, we can treat the idea of finding a mapping that generalizes well more formally. What we seek to achieve is to minimize the *expected risk* of the hypothesis $\hat{y}_\theta$:[1]

$$R(\theta) = \mathop{\mathbb{E}}_{p(x,y)} \big[C(x, y, \theta)\big] = \sum_{\mathcal{X}\times\mathcal{Y}} C(x, y, \theta) p(x, y)\,,$$

where the expectation is taken with respect to the unknown data-generating distribution $p(x, y)$.[2] Since we do not have access to $p(X, Y)$, we cannot minimize the expected risk directly. However, assume that we have access to a finite sample $\mathcal{D} = \{(x^{(j)}, y^{(j)})\} \subset (\mathcal{X} \times \mathcal{Y})^m$ of m instances assumed to be *identically and independently distributed* (iid) according to $p(X, Y)$. With this data set, we can perform *empirical risk minimization*, where we instead seek to minimize the empirical risk of θ over the sample $\mathcal{D}$:

$$\hat{R}(\theta; \mathcal{D}) = \hat{\mathbb{E}}_{\mathcal{D}} \big[C(x, y, \theta)\big] = \frac{1}{m} \sum_{j=1}^{m} C(x^{(j)}, y^{(j)}, \theta)\,.$$

In line with the learning-theoretical principle of structural risk minimization (Vapnik and Chervonenkis, 1971; Vapnik, 1998), we add a *regularizer* $\Omega\colon \Theta \to \mathbb{R}_+$ that controls the *capacity* — the ability to fit the data with complex hypotheses — of the discriminant function $s_\theta(\cdot)$. This results in the *regularized empirical risk*

$$\hat{R}_\Omega(\theta; \mathcal{D}) = \hat{R}(\theta; \mathcal{D}) + \lambda\Omega(\theta)\,, \tag{4.1}$$

where the hyper-parameter λ controls the trade-off between "goodness-of-fit" against function complexity. Regularization is important, because if we allow

[1] Alternative learning frameworks exist. For example, *learning with expert advice* drops the assumption of a data-generating distribution and the notion of expected risk is replaced with that of the *regret* of the learner (Cesa-Bianchi and Lugosi, 2006).

[2] Here we have assumed that both $\mathcal{X}$ and $\mathcal{Y}$ are sets of discrete structures. If either of these were continuous, the summation would be replaced by the corresponding integral.

arbitrary complex discriminant functions, we could end up with a mapping that achieves zero risk on the training set, while performing terribly on any other data. In particular, this is an issue when the size of the training set is small. There are learning-theoretical results that specify how the generalization ability depends on the capacity of the hypothesis class and the sample size (Vapnik, 1998). These results are nearly all based on the assumption that we are learning from an unbiased sample from $p(X, Y)$ (Haussler, 1992). Of course, this is rarely a valid assumption, though in many cases we can come reasonably close. However, in cross-domain or cross-lingual transfer, this assumption is clearly violated, as discussed in chapter 7.

4.1.2 Cost Functions and Evaluation Measures

The most straight-forward sensible cost function is the 0/1 cost $\delta\colon \mathcal{Y} \times \mathcal{Y} \to \{0, 1\}$, which measures whether two outputs $y, y' \in \mathcal{Y}$ are equivalent or not, where $\mathcal{Y} \subset \{0, 1\}^{|\mathcal{I}|}$ (see section 3.2 for definitions of the various index sets):[3]

$$\delta(y, y') \;= \mathbb{1}\left[y = y'\right] = \mathbb{1}\left[y(i) = y'(i),\ \forall i \in \mathcal{I}\right]. \tag{4.2}$$

Another natural and more informative cost function can be formed by assigning a cost to each individual substructure. The normalized Hamming cost $\Delta\colon \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ between y and y' is defined as the percentage of active substructures on which y and y' differ:

$$\Delta(y, y') \;= \frac{1}{n} \left|\{\, i \in \mathcal{I} : y(i) \neq y'(i) \,\}\right|, \tag{4.3}$$

where $n = \left|\{\, i \in \mathcal{I} : y(i) = 1 \,\}\right|$ is the number of active substructures in $\mathcal{I}$. Note that this assumes that the number of active substructures is equal in y and y'. For all structures studied in this dissertation n is by definition constant across every $y \in \mathcal{Y}$.

From these cost functions we define the evaluation measures that we use to measure the performance of a learned predictor. The *exact match score* of a prediction y' with respect to the correct output y is defined as the reciprocal of the 0/1 cost:

$$\text{match}(y, y') \;= 1 - \delta(y, y'). \tag{4.4}$$

Similarly, the *accuracy* of the prediction is defined as the reciprocal of the normalized Hamming cost:

$$\text{accuracy}(y, y') \;= 1 - \Delta(y, y'). \tag{4.5}$$

[3]Note that this assumes that there is a one-to-one correspondence between indices and output structure. For transition-based parsing models this is not necessarily the case, since different transition sequences may correspond to the same dependency tree. However, assuming a mapping form transition sequences to an arc-factorization, the definition remains valid.

The exact match score and accuracy of a set of predictions is simply the exact match score and accuracy averaged over all the predictions in the set. These averages computed over a separate test set is used to evaluate the performance of a learned predictor. Averaging in this way, that is, computing the accuracy per instance and then averaging these accuracies over a set of instances, is referred to as *macro-averaging*. We may also treat the whole test set prediction as one large structured output with an additional index that picks up individual examples from the test set and then use the above cost functions with this augmented index set. This is known as *micro-averaging*.

For sequence-labeling tasks with atomic labels, such as part-of-speech tagging, we use eq. (4.4) and eq. (4.5) for evaluation, taking $\mathcal{I}_{\text{seq}}$ as the index set. For dependency parsing we use two different evaluation measures, which correspond to two different index sets. The *labeled attachment score* (LAS) is defined as the percentage of tokens with correctly assigned labeled head attachment (that is, taking $\mathcal{I}_{\text{arc}}$ as the index set), while the *unlabeled attachment score* (UAS) is defined as the percentage of tokens with correctly assigned heads, ignoring labels (that is, taking $\mathcal{I}_{\text{uarc}}$ as the index set).

For named-entity recognition, these cost functions and evaluation measures are somewhat problematic. The reason is that there are many more tokens that are not part of a named entity, compared to the number of tokens that are. Therefore, by simply classifying all tokens as not being part of a named entity, we could achieve a high accuracy, although the resulting predictor would be useless. This is partly a problem for sentiment analysis as well. For example, in a positive review, we are much more likely to find sentences that express a positive sentiment compared to those that express a negative sentiment. Instead, we use the F_β measure (van Rijsbergen, 1979), which is more appropriate in the face of imbalanced classes:

$$F_\beta(y, y') = (1 + \beta^2)\frac{\text{precision}(y, y') \cdot \text{recall}(y, y')}{\beta^2 \cdot \text{precision}(y, y') + \text{recall}(y, y')}\,.$$

When applied to named entities, we follow the convention of the CoNLL shared tasks on multilingual named entity recognition (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) and define precision(y, y') as the percentage of named entities in y' that are correct with respect to those in y and recall(y, y') as the percentage of gold-standard named entities that are correctly predicted (again, taking y as the gold-standard and y' as the prediction). For a named entity to be considered to be correctly predicted, we require that all tokens that are part of the named entity are correctly predicted (including the entity type). For sentiment analysis, we use accuracy as well as precision, recall and F_β for evaluation, taking $\mathcal{I}_{\text{sent}}$ as the index set.

In this dissertation, precision and recall are weighted equally, which is achieved by letting $\beta = 1$. This leads to the F_1 measure, which is the most

commonly used evaluation measure for information extraction tasks:

$$F_1(y, y') = 2 \cdot \frac{\text{precision}(y, y') \cdot \text{recall}(y, y')}{\text{precision}(y, y') + \text{recall}(y, y')} .$$

Because the precision, recall and F_1 measures do not decompose linearly over the substructures, the corresponding cost functions are more difficult to minimize, compared to the 0/1 cost and Hamming cost. It is possible to devise cost functions such that minimizing eq. (4.1) corresponds to maximizing the empirical precision, recall and F_β measures (Joachims, 2005; Dembczynski et al., 2011). Methods for learning with arbitrary non-decomposable cost functions were recently proposed by Tarlow and Zemel (2012) and Pletscher and Kohli (2012). Since our focus is on learning with different types of incomplete information, to which the choice of cost function is largely orthogonal, we only make use of the 0/1 cost in this dissertation — or rather surrogates for it as described in the next section. When evaluating the learned predictor, we of course use the task specific evaluation measures.

4.1.3 Surrogate Loss Functions

Unfortunately, even the linearly decomposable cost functions above are difficult to optimize (Taskar et al., 2003; Tsochantaridis et al., 2005; McAllester et al., 2010). In fact, the problem of minimizing 0/1 cost directly is known to be NP-hard, except for the case when the training set is separable, which we cannot assume to hold in general. The difficulty is due to the fact that we are working with discrete outputs, which means that the cost function is highly discontinuous. The cost at a particular point $C(x, y, \theta)$ thus provides very little information on how to change the parameters θ to achieve lower risk. A standard way to circumvent this issue is to instead use a *surrogate loss* function $L\colon \mathcal{X} \times \mathcal{Y} \times \Theta \to \mathbb{R}_+$, where $L(x, y, \theta)$ is a continuous convex upper-bound on the cost $C(x, y, \theta)$. A further assumption is that L is at least subdifferentiable with respect to θ. Since the sum of convex functions is convex (Boyd and Vandenberghe, 2004), if we replace C with L in eq. (4.1), and assume a convex regularizer, we obtain a convex upper-bound on the empirical risk:

$$J(\theta; \mathcal{D}) = \frac{1}{m} \sum_{j=1}^{m} L(x^{(j)}, y^{(j)}, \theta) + \lambda \Omega(\theta) . \tag{4.6}$$

Learning then amounts to finding the parameter vector $\hat{\theta}$ that minimizes this objective, given some data set $\mathcal{D}$:

$$\hat{\theta} = \arg\min_{\theta} J(\theta; \mathcal{D}) .$$

The convexity and sub-differentiability of eq. (4.6) allows us to use simple gradient-based methods for learning, as discussed in section 4.2.

Surrogate losses for 0/1 cost

The simplest convex surrogate loss is the *perceptron loss*

$$L_{\text{per}}(x, y, \theta) \;=\; -\left(s_\theta(x, y) - \max_{y' \in \mathcal{Y}(x)} s_\theta(x, y')\right) \;=\; -\left(s_\theta(x, y) - \hat{s}_\theta(x)\right). \quad (4.7)$$

This corresponds to the loss optimized by the perceptron algorithm (Rosenblatt, 1958) in the case of binary outputs, and by the structured perceptron algorithm (Collins, 2002) in the case of structured outputs.

Replacing the max in the perceptron loss with the "soft-max" (log-sum-exp), yields the *log loss*:

$$L_{\log}(x, y, \theta) \;=\; -\left(s_\theta(x, y) - \log Z_\theta(x)\right), \quad (4.8)$$

where the partition function

$$Z_\theta(x) \;=\; \sum_{y' \in \mathcal{Y}(x)} \exp\left\{s_\theta(x, y')\right\}.$$

Whereas the max is only subdifferentiable, the soft-max is a differentiable upper bound on the max. This loss is of particular interest to us as it is equivalent to the negative log-likelihood $-\log p_\theta(y \mid x)$ of a globally normalized conditional model; see section 3.4.1. Minimizing the log loss is thus equivalent to maximizing the (log-)likelihood of the training data. For a detailed theoretical analysis of the log loss, see Cesa-Bianchi and Lugosi (2006, ch. 9).

When applied to a locally normalized log-linear model (see section 3.4.2), the log loss (the negative joint log-likelihood) decomposes into a sum of log losses over local factors. For example, in the case of a kth-order Markov model, we have that

$$\begin{aligned} -\log p_\beta(x, y) &= -\left(\sum_{i=1}^{|x|} \log p_\beta(x_i \mid y_i) + \log p_\beta(y_i \mid y_{i-k:i-1})\right) \\ &= \sum_{i=1}^{|x|} L_{\log}(x_i, y_i, \beta) + L_{\log}(y_i, y_{i-k:i-1}, \beta). \end{aligned}$$

Neither the perceptron loss nor the log loss can take cost functions other than 0/1 cost into account, which somewhat limits their applicability. The perceptron loss is particularly problematic, because with this loss the regularization term in eq. (4.6) becomes vacuous. The reason is that since $\hat{y}_\theta(x) = \hat{y}_{\alpha \cdot \theta}(x)$ for all $\alpha > 0$, the regularization has no effect on the empirical risk; we can always uniformly rescale the parameters by α to minimize the regularizer and still get the same predictions. This is not a problem for the log loss, since the partition function is scale-sensitive in this respect and regularization thus has an effect on the empirical risk. In section 4.2 we discuss a way to partially circumvent this problem for the perceptron loss, which involves letting θ be the average of a collection of parameter vectors.

Margin-based surrogate losses

There are multiple ways to incorporate arbitrary decomposable cost functions into a surrogate loss function (Taskar et al., 2003; Tsochantaridis et al., 2005). A particularly simple way to achieve this in the regularized empirical risk minimization framework is to rescale the *margin* by the cost function, so that outputs with a high cost are penalized more compared to outputs with a lower cost and thus pushed further away from the decision boundary defined by $\hat{y}_\theta(\cdot)$. The margin of an example (x, y) with respect to the parameters θ is defined as the difference in score between the correct label y and the best incorrect label $\hat{y}(x)$:

$$m(x, y, \theta) \; = s_\theta(x, y) - s_\theta(x, \hat{y}(x)) = s_\theta(x, y) - \hat{s}_\theta(x)\,,$$

where

$$\hat{y}(x) \; = \; \operatorname*{arg\,max}_{y' \in \mathcal{Y}(x) \setminus \{y\}} \; s_\theta(x, y')\,.$$

Consequently,

$$\hat{s}_\theta(x) = \max_{y' \in \mathcal{Y}(x) \setminus \{y\}} s_\theta(x, y')\,.$$

The margin is a measure of how well the score function $s_\theta(x, \cdot)$ predicts the correct label y for a given θ. The margin is positive only if $\hat{y}_\theta(x) = y$, that is, if x is classified correctly. The magnitude of a positive margin can be interpreted as a measure of how "safe" the classification is. If the margin is small, a small random perturbation of $\Phi(x, \cdot)$ or θ runs the risk of changing $\hat{y}_\theta(x)$ to an incorrect classification, while if the margin is large, larger perturbations are required for this to occur. This intuitive notion of variance reduction can be formulated more formally and there is a large body of learning theoretic results in which the margin concept plays a fundamental role in the derivation of generalization bounds (Vapnik, 1998).

In *margin-based* learning methods, constraints are placed on the margins of the examples in the training set, such that a minimal margin of separation is required. By linearly scaling these margin constraints by the value of the cost function, we can bias the discriminant function to push away outputs with high cost from the outputs with lower cost. At the optimal θ, the following constraint for all pairs $(x, y) \in \mathcal{D}$ is thus enforced:

$$m(x, y, \theta) \; \geq C(y, \hat{y}_\theta(x)) \Leftrightarrow s_\theta(x, y) - \left[\hat{s}_\theta(x) + C(y, \hat{y}_\theta(x))\right] \geq 0\,.$$

Minimizing the regularized empirical risk subject to these constraints is equivalent to plugging in the *margin-scaled hinge loss*

$$L_{\text{hinge}}(x, y, \theta) \; = -\left(s_\theta(x, y) - \hat{s}_\theta(x, y, C)\right) \tag{4.9}$$

in eq. (4.6), where

$$\hat{s}_\theta(x, y, C) \; = \max_{y' \in \mathcal{Y}(x)} \left[s_\theta(x, y') + C(y, y')\right]\,. \tag{4.10}$$

In case of binary outputs, the hinge loss corresponds to a binary support vector machine (SVM; Cortes and Vapnik, 1995), while in the case of structured outputs, it corresponds to the loss optimized in a max-margin Markov network (M^3N; Taskar et al., 2003), or equivalently the margin-rescaled version of the structural SVM (Tsochantaridis et al., 2005). A restricted variant of this formulation, where the cost function is fixed to 0/1 cost, was proposed by Altun et al. (2003). The problem of solving the max in eq. (4.10) is often referred to as the *loss augmented* inference problem.

While margin-based loss function may sometimes be preferable, we only use the perceptron loss and the log loss in this dissertation. There are two reasons for this choice. First, in practice the above loss functions all tend to perform similarly for the natural language processing tasks that we study, with the caveat that we need to use an averaging trick to achieve good generalization performance in place of regularization with the perceptron loss, as discussed above. Second, some of our methods rely on probabilistic quantities and the log loss naturally equips the model with this capability, which is not true for either the perceptron loss or the hinge loss. If cost sensitivity is a major issue, Pletscher et al. (2010), Hazan and Urtasun (2010) and Gimpel and Smith (2010) independently proposed a cost-sensitive version of log loss, which is derived by replacing the max in eq. (4.9) by the soft-max. All our methods could easily be adapted to use this loss function instead, as long as the cost function is decomposable according to the model structure. We will return to margin-based loss functions when discussing related work for learning with partial information in chapter 5.

4.1.4 Regularizers

With the loss function in place, all that is left to complete eq. (4.6) is to specify the regularizer. Depending on which class of functions the hypothesis class belongs to, different regularizers can be appropriate. When restricted to linear functions, as is the case here, the two most commonly used regularizers are the ℓ_2-regulariser

$$\Omega_{\ell_2}(\theta) \;=\; \|\theta\|_2^2 = \theta^T\theta \tag{4.11}$$

and the ℓ_1-regularizer

$$\Omega_{\ell_1}(\theta) \;=\; \|\theta\|_1 = \sum_i |\theta_i| \,.$$

These are both instances of the more general ℓ_p-regularizer

$$\Omega_{\ell_p}(\theta) \;=\; \|\theta\|_p^p = \sum_i |\theta_i|^p \,.$$

Note that the ℓ_p-norm is non-convex for $p < 1$. The intuition behind these regularizers is that by keeping the elements of θ small, we limit the expressivity of the corresponding linear discriminant function $s_\theta(\cdot)$ and in turn of the

hypothesis class $\hat{y}_\theta(\cdot)$. Since the ℓ_2-regularizer is both convex and continuous, the minimization of the regularized empirical risk is simplified, and it is therefore the most commonly used regularizer in practice. However, if sparse solutions are preferred, that is, solutions in which as few elements of θ as possible are non-zero, the ℓ_1-regularizer is preferable. While the ℓ_1-regularizer is convex, it is not continuous everywhere, which makes is slightly more difficult to optimize.

When combined with the log loss, the ℓ_1- and ℓ_2-regularizers have natural interpretations as Bayesian prior distributions on the parameters. Using the ℓ_2-regularizer with the log loss corresponds to *maximum a posteriori* (MAP) estimation in an exponential family model with a diagonal gaussian prior distribution over the parameters (Chen and Rosenfeld, 1999), whereas the ℓ_1-regularizer corresponds to MAP estimation with a Laplace distribution as the prior distribution (Williams, 1995). These regularizers go by many names. Another term for ℓ_2-regularization, when used for regression with square loss, is *Tikhonov* regularization (Tikhonov and Arsenin, 1977) and in the case of ℓ_2-regularized multiple linear regression it is known as *ridge regression* (Hoerl and Kennard, 1970). In the statistics community, the ℓ_1-regularizer is also known as the *Lasso* when used with square loss (Tibshirani, 1994), while a combination of ℓ_2- and ℓ_1- regularization in MAP estimation is known as the *Elastic Net* (Zou and Hastie, 2005).

4.2 Gradient-Based Optimization

As discussed, when a convex loss-function is combined with a convex regularizer, the regularized empirical risk in eq. (4.6) is also a convex function. A convex differentiable function $f(v)$ has a unique minimal solution $\hat{v} = \min_\theta f(v)$. In particular, if $\frac{\partial}{\partial v} f(v) = 0$, for some v, then it is guaranteed that $v = \hat{v}$, where $\frac{\partial}{\partial v} f(v)$ is the vector of first-order partial derivatives of f evaluated at v (Boyd and Vandenberghe, 2004). For simple cases, such as a linear discriminant function with the squared regression loss, the minimum has a closed form solution. Unfortunately with the non-linear loss functions and regularizers that we consider, the regularized empirical risk has no closed-form solutions, so we revert to iterative gradient-based optimization methods. There are many different gradient-based methods for minimizing convex differentiable functions. The idea behind all is to exploit the curvature of the function, represented by its partial derivatives, by iteratively modifying v such that $f(v)$ eventually shrinks to its minimum. Of course, we want the search for the minimum to converge as fast as possible; however, there may be a trade-off between algorithmic and computational complexity on the one hand, and convergence rate on the other. Commonly, only the first-order partial derivative is used to represent the curvature, while some methods make use of (approximations of) the second-order partial derivatives as well.

4.2.1 Online versus Batch Optimization

In this dissertation, we make use of two standard optimization algorithms for learning. The first, *stochastic gradient descent* (SGD; Bottou and Bousquet, 2007), is a first-order *online* learning method. The second, L-BFGS (Liu and Nocedal, 1989), is a *batch* learning method, which exploits approximate second-order information to improve convergence.

An online learning algorithm processes one example from the training set at a time, while possibly making multiple passes over the data. This is in contrast to batch learning algorithms, which process the whole training set as a single unit at each iteration. Online methods have several advantages (Bottou and Bousquet, 2007). First, by processing each example individually, the memory requirements are typically small, compared to batch methods. Second, online methods typically enjoy faster convergence rates than batch methods, especially in early stages of the optimization (Agarwal et al., 2011). Third, since examples are processed individually, online methods have the potential to learn from infinite streams of data and to incrementally adapt to drift in the data-generating distribution, which is not possible with batch methods (Shalev-Shwartz, 2012).

Thanks to their smaller memory requirements and often faster initial convergence rates, interest in online algorithms has increased as training sets are becoming ever larger (Zhang, 2004; Bottou and Bousquet, 2007). On the other hand, online methods also have some disadvantages. Primarily, although the rate of convergence is typically fast initially, after a sufficient number of iterations batch methods can often reach somewhat better solutions. Mini-batch algorithms try to balance this trade-off by aggregating the gradient over a small subset of examples before performing an update to the parameters. This approach also has the benefit that the computation of the gradient can be trivially parallelized over as many processors as there are examples in each mini-batch (Gimpel et al., 2010). Purely online algorithms that process one instance at a time are slightly more difficult to parallelize, but several methods for accomplishing this has been proposed, for example, by McDonald et al. (2010) and by Niu et al. (2011). Of course, each iteration of a full batch algorithm is fully parallelizable, as the contribution to the gradient of each instance can be computed on a separate processor. Other more involved strategies have also been considered (Mann et al., 2009). The data sets considered in this dissertation for supervised and partially supervised learning are all small enough, so that we do not need to employ parallel learning algorithms. However, in chapter 8, we use a parallel algorithm to induce word clusters from large amounts of raw text.

Next, we describe two variants of the SGD algorithm, but first let us derive the gradients of the loss functions and the regularizer that we use in the dissertation. We refer the reader to Liu and Nocedal (1989) for an in depth description of the L-BFGS algorithm.

4.2.2 Gradients of Loss Functions and Regularizers

We will use the gradients of the perceptron loss, the log loss and the ℓ_2-regularizer. We give the *stochastic gradient* for an example (x, y) here, that is, the part of the gradient contributed by that example. The gradient used in batch methods is simply the sum of the stochastic gradients taken over all examples in the training set.

Recall the perceptron loss $L_{\text{per}}(x, y, \theta) = -(s_\theta(x, y) - \hat{s}_\theta(x))$. The stochastic gradient of this loss with respect to θ is

$$\begin{aligned}\frac{\partial}{\partial\theta} L_{per}(x, y, \theta) &= -\left(\frac{\partial}{\partial\theta} s_\theta(x, y) - \frac{\partial}{\partial\theta}\hat{s}_\theta(x)\right) \\ &= -\left(\frac{\partial}{\partial\theta}\theta^\top\Phi(x, y) - \frac{\partial}{\partial\theta}\theta^\top\Phi(x, \hat{y}_\theta)\right) \\ &= -\left(\Phi(x, y) - \Phi(x, \hat{y}_\theta)\right).\end{aligned}$$

In other words, the gradient is the negative difference between the feature vector of the correct output and the feature vector of the output predicted by the current parameter vector θ. Taking a step in the negative direction of the gradient thereby alters the parameters such that the same example would be more likely to be correctly classified after the update.

If the model predicts the correct example, the gradient vanishes. Using perceptron loss with SGD is thus equivalent to the perceptron algorithm. Note that since the gradient vanishes, when all examples are classified correctly, no further updates will occur. This is in contrast to margin-based loss functions and the log loss.

The log loss is defined as $L_{\log}(x, y, \theta) = -(s_\theta(x, y) - \log Z_\theta(x))$, where $Z_\theta(x) = \sum_{y' \in \mathcal{Y}(x)} \exp\left\{s_\theta(x, y')\right\}$. The stochastic gradient of this loss with respect to θ can easily be derived by application of the chain rule to $\frac{\partial}{\partial\theta}\log Z_\theta(x)$. Recall that for any differentiable function $f\colon v \to \mathbb{R}$ it holds that

$$\frac{\partial}{\partial v}\log f(v) = \frac{1}{f(v)}\frac{\partial}{\partial v} f(v) \quad \text{and} \quad \frac{\partial}{\partial v}\exp\left\{f(v)\right\} = \exp\left\{f(v)\right\} \cdot \frac{\partial}{\partial v} f(v).$$

Applied to the gradient of the log-partition function, we find that

$$\begin{aligned}\frac{\partial}{\partial\theta}\log Z_\theta(x) &= \frac{1}{Z_\theta(x)} \sum_{y' \in \mathcal{Y}(x)} \exp\left\{s_\theta(x, y')\right\} \frac{\partial}{\partial\theta} s_\theta(x, y') \\ &= \frac{1}{Z_\theta(x)} \sum_{y' \in \mathcal{Y}(x)} \exp\left\{s_\theta(x, y')\right\} \cdot \Phi(x, y') \\ &= \sum_{y' \in \mathcal{Y}(x)} p_\theta(y' \mid x) \cdot \Phi(x, y') \\ &= \mathop{\mathbb{E}}_{p_\theta(y' \mid x)}\left[\Phi(x, y')\right].\end{aligned}$$

Algorithm 1 Stochastic Gradient Descent

$\mathcal{D}$: Training set
L: Loss function
Ω: Regularizer
λ: Regularization trade-off hyper-parameter
T: Number of epochs
η: Sequence of step sizes $\eta_1, \eta_2, \ldots$
procedure SGD($\mathcal{D}, L, \Omega, \lambda, T, \eta$)
    $\theta \leftarrow \mathbf{0}$
    for $t = 1, 2 \ldots T$ **do**
        $(x, y) \sim \mathcal{D}$ ▷ Pick a random example
        $g \leftarrow \frac{\partial}{\partial \theta} L(x, y, \theta) + \frac{\lambda}{|\mathcal{D}|} \frac{\partial}{\partial \theta} \Omega(\theta)$ ▷ Compute gradient
        $\theta \leftarrow \theta - \eta_t g$ ▷ Take gradient step
    end for
    return θ ▷ Return final parameters
end procedure

Consequently, the gradient of the log loss is given by

$$\frac{\partial}{\partial \theta} L_{\log}(x, y, \theta) = -\left(\Phi(x, y) - \mathop{\mathbb{E}}_{p_\theta(y'|x)} \left[\Phi(x, y')\right]\right). \tag{4.12}$$

In other words, the gradient is the difference between the observed feature counts $\Phi(x, y)$ and the expected feature counts $\mathbb{E}_{p_\theta(y'|x)} \left[\Phi(x, y')\right]$. Hence, at the optimum $\hat{\theta}$, where the gradient is zero, the model exactly matches the empirical distribution (in the sense that its moments match the empirical moments). In a locally normalized model, the gradient instead takes the form of a sum of the gradients of the log losses over local factors.

Finally, the gradient of the ℓ_2-regularizer is

$$\frac{\partial}{\partial \theta} \Omega_{\ell_2}(\theta) = \frac{1}{2}\theta. \tag{4.13}$$

While the ℓ_2-regularizer is trivially optimized with gradient decent methods, the ℓ_1-regularizer is more difficult to optimize, partly due to it not being differentiable at $\theta = 0$ (Andrew and Gao, 2007; Duchi et al., 2008; Tsuruoka et al., 2009).

Stochastic gradient descent

A wide variety of online learning methods have been proposed in the literature (Kivinen and Warmuth, 1997; Crammer et al., 2006; Hazan et al., 2007; Shalev-Shwartz et al., 2011). Stochastic gradient descent (SGD; Robbins and Monro, 1951) is a general online optimization algorithm applicable to any unconstrained optimization problem for which the gradient of the objective

function can be decomposed over individual examples.[4] The basic SGD algorithm is very easy to implement. First, the parameters are initialized to some random position of the search space, typically at $\theta = 0$. The parameters are then iteratively updated in the direction opposite to the gradient of the minimization function, evaluated at each example. The general form of the algorithm, for minimization problems, is shown in algorithm 1. Despite its simplicity and the fact that it is quite a poor *optimization* algorithm (Bottou, 2004), it has proved to work remarkably well for solving *learning* problems, as discussed extensively by Bottou and Bousquet (2007).

One issue with the SGD algorithm is that its convergence rate, that is, the rate at which it approaches the minimum, is sensitive to the *learning rate* sequence $\eta_1, \eta_2, \ldots$. A common choice is to use $\eta_t \propto \eta t^{-1} < 1$, for some constant η (Robbins and Monro, 1951). Other learning rate sequences may provide faster convergence, but for simplicity we have chosen to stick with a constant learning rate $\eta_t = \eta$. The generalization error will often vary substantially with the specific value of η. We therefore use cross-validation to select the value of $\eta \in \mathcal{E}$ that gives the best estimated generalization error, using, for example, $\mathcal{E} = \{0.00001, 0.0001, 0.001, 0.01, 0.1\}$. While this simple scheme turns out to work quite well in practice, more complex approaches have been proposed. In particular, there are methods that adapt the step size automatically during learning, for example, *stochastic meta-descent* (Schraudolph, 1999; Vishwanathan et al., 2006), *AdaGrad* (Duchi et al., 2011) and the recent adaptive approach of Schaul et al. (2012).

The basic SGD algorithm as described here only applies to differentiable objective functions. For objectives that are only subdifferentiable — such as the hinge loss — *stochastic subgradient descent* can be used instead (Ratliff et al., 2007). The primary difference from SGD is that rather than taking a step in the opposite of the gradient direction, a subgradient step is instead taken. There are also variants of SGD that can be used for learning in reproducing-kernel Hilbert spaces (Kivinen et al., 2004). A kernelized version of the perceptron algorithm was introduced by Freund and Schapire (1999).

Averaged stochastic gradient descent

As discussed in section 4.1.3, the perceptron loss makes the regularization term in the regularized empirical risk vacuous. However, by a slight change to the algorithm, SGD with the perceptron loss can be turned into a large-margin classifier, which is another way of controlling the capacity of the discriminant function. While the standard SGD algorithm returns the final parameter vector, the *averaged stochastic gradient descent* algorithm, shown in algorithm 2,[5] returns the average of the parameter vectors from each iteration (Ruppert, 1988; Polyak and Juditsky, 1992). Another variant of this algorithm

[4] It can also be applied to non-convex problems, as discussed in chapter 5. However, in the non-convex case, the solution is only guaranteed to be locally optimal.

[5] The cumulative averaging trick in algorithm 2 is described by, for example, Daumé III (2006).

Algorithm 2 Averaged Stochastic Gradient Descent

$\mathcal{D}$: Training set
L: Loss function
Ω: Regularizer
λ: Regularization trade-off hyper-parameter
T: Number of epochs
η: Sequence of step sizes $\eta_1, \eta_2, \ldots$
procedure AVERAGEDSGD($\mathcal{D}, L, \Omega, \lambda, T, \eta$)
    $\theta \leftarrow \mathbf{0}$
    $\bar{\theta} \leftarrow \mathbf{0}$
    for $t = 1, 2 \ldots T$ **do**
        $(x, y) \sim \mathcal{D}$ ▷ Pick a random example
        $g \leftarrow \frac{\partial}{\partial \theta} L(x, y, \theta) + \frac{\lambda}{|\mathcal{D}|} \frac{\partial}{\partial \theta} \Omega(\theta)$ ▷ Compute gradient
        $\theta \leftarrow \theta - \eta_t g$ ▷ Take gradient step
        $\bar{\theta} \leftarrow \bar{\theta} - t \eta_t g$ ▷ Aggregate gradient step
    end for
    return $\theta - \frac{1}{T} \bar{\theta}$ ▷ Return averaged parameters
end procedure

based on voting rather than averaging was proposed by Freund and Schapire (1999), for the special case of the perceptron loss. Empirically, using averaging is critical to achieve good generalization ability with the perceptron loss (Collins, 2002). However, for loss functions that are scale sensitive, the benefit of averaging is less clear. In some cases using averaging can be shown to give better convergence rate, yet in others it can be shown to give worse (Rakhlin et al., 2012). We therefore only use averaged SGD together with the perceptron loss. When learning with log loss, we revert to the standard SGD algorithm.

4.2.3 Tricks of the Trade

There are some simple tricks that facilitate a more efficient implementation of learning algorithms. First, we can exploit sparseness in the gradient vectors. Second, the form of the ℓ_2-regularizer allows for a simple parameter rescaling. Third, a large part of the joint feature function can often be efficiently decomposed into an input and an output part. Fourth, and finally, we can use *feature hashing* to perform fast computation of feature vectors and to fix the dimensionality of the feature space.

Sparse updates

While the parameter vector is generally dense, the feature vectors are often very sparse. This sparseness is mainly an effect of the use of binarized lexical

features and of the large number of possible substructures and combinations of substructures in $\mathcal{Y}(x)$. When implementing gradient-based learning with gradients of the form $\Phi(x, y) - \Phi(x, \hat{y}_\theta)$, such as with the perceptron loss and the hinge loss, we can use an efficient sparse representation of the difference vector. Unfortunately, this is not true for the log loss, for which the gradient is much less sparse, due to the derivative of the soft-max.

Parameter rescaling with the ℓ_2-regularizer
As can be seen from *eq*. (4.13), the gradient of the ℓ_2-regularizer is dense. However, note that this gradient simply corresponds to a uniform rescaling of the parameters by the scalar $\left(1 - \frac{\eta_t \lambda}{|\mathcal{D}|}\right)$. This fact can be used to implement the gradient update very efficiently, which is of importance especially with SGD where the gradient of the regularizer is included in the update for each example. The trick is to represent the parameter vector as the combination of a dense vector θ' and a scalar α, that is, by letting $\theta = \alpha\theta'$.[6] The ℓ_2-regularizer gradient step can thus be implemented simply by rescaling α, while the sparse part of the gradient only affects θ'. By factoring lines 11-12 of algorithm 1 (analogously, lines 12-14 of algorithm 2) into

$$\theta \leftarrow \theta - \eta_t \frac{\partial}{\partial \theta} L(x, y, \theta) - \frac{\eta_t \lambda}{|\mathcal{D}|}\theta = (1 - \frac{\eta_t \lambda}{|\mathcal{D}|})\alpha\theta' - \eta_t \frac{\partial}{\partial \theta} L(x, y, \alpha\theta')\,,$$

we only need to perform the following updates at each iteration:

$$\alpha \leftarrow \alpha - \frac{\eta_t \lambda}{|\mathcal{D}|}$$

and

$$\theta' \leftarrow \theta' - \eta_t \frac{\partial}{\partial \theta}(x, y, \alpha\theta')\,.$$

Decomposing joint feature functions
Extraction of feature vectors — that is, the mapping of inputs and outputs to a collection of elements of the feature space — often consumes a substantial part of the runtime of a linguistic structure prediction system. Joint feature functions enable encoding of statistical dependencies between input and substructures of the output. However, for most feature templates the input part of the feature function remains invariant, while some subset of the structured index $i \in \mathcal{I}(x)$ is varied. In such cases it makes practical sense to extract the input features once and then combine them with each assignment of the relevant output substructures. Recall the problem of sequence labeling with a first-order Markov assumption, where we have the sequence index set

$$\mathcal{I}^1_{\text{seq}}(x) = \{\, (i, t_{i-1}t_i) : i \in [1, |x|],\ t_{i-1} \in \mathcal{T},\ t_i \in \mathcal{T} \,\}\,,$$

[6]This trick is used, for example, in Leon Bottou's highly efficient implementation of SGD, available at http://leon.bottou.org/projects/sgd — January 15, 2013.

where i is the position in the sequence, t_i is the tag at position i, and t_{i-1} is the tag at position $i-1$. The joint feature function $\Phi(x, y)$ can in this case be decomposed into feature functions over the input $\phi(x, i)$ as well as over pairs of active output substructures $\psi(y_{,}y_{i-1})$; see Altun et al. (2003):[7]

$$\Phi(x, y) = \sum_{i=1}^{n} \phi(x, i, y_i, y_{i-1}) = \sum_{i=1}^{n} \phi(x, i) \otimes \psi(y_i, y_{i-1}),$$

where $y_i \in \mathcal{T}$ and $y_{i-1} \in \mathcal{T}$ are the labels at position i and $i-1$, respectively, and the cross-product operator $\otimes$ maps each combination of input feature and output feature to a unique element in $\Phi(x, y)$. The upshot of this decomposition is that the cross-product operator can be implemented very efficiently.

Feature hashing

With joint feature functions, the number of unique features may grow very large. This is a potential problem when the amount of internal memory is limited. Feature hashing is a simple trick to circumvent this problem (Weinberger et al., 2009). Assume that we have an original feature function $\Phi: \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$. Let $h: \mathbb{N}^+ \to [1, n]$ be a hash function and let $h^{-1}(i) \subseteq [1, d]$ be the set of integers such that $j \in h^{-1}(i)$ if and only if $h(j) = i$. We now use this hash function to map the index of each feature in $\Phi(x, y)$ to its corresponding index in a hashed feature vector $\Phi'(x, y)$, by letting $\Phi'_i(x, y) = \sum_{j \in h^{-1}(i)} \Phi_j(x, y)$. Each feature in $\Phi'(x, y)$ may thus correspond to multiple features in $\Phi(x, y)$. Given a hash function with good collision properties, we can expect that the subset of features mapped to any particular index in $\Phi'(x, y)$ is small, and composed of elements drawn at random from $\Phi(x, y)$.

[7] Note that $\Phi(x, y)$ might have a smaller number of components that do not factor in this way. For example, we could have a joint feature that couples (x_i, y_{i-1}, y_i). However, these features will typically be few compared to the number of features that decompose into an input part and and output part.

Part II:
Learning with Incomplete Supervision

5. Learning with Incomplete Supervision

This chapter provides an overview of various learning scenarios in which only incomplete supervision, or no supervision, is available to the learner. After a non-technical overview, we turn to a more detailed technical description of structured latent variable models, which constitute the key modeling tool in subsequent chapters. While these models are similar to the structured prediction models in chapter 3, the loss functions from chapter 4 need to be generalized to enable learning with no, partial and ambiguous supervision.

5.1 Types of Supervision

Before describing the methods employed in subsequent chapters in more technical detail, we provide a brief overview of different types of supervision that have been proposed and studied in the literature on structured prediction and classification. As in previous chapters, $\mathcal{X}$ denotes the input space and $\mathcal{Y}$ denotes the output space. In some of these learning scenarios, an additional space of *latent* (hidden) structure $\mathcal{Z}$ is added to the input and output spaces. The semantics of these spaces are that inputs are always observed and outputs are observed during learning, whereas latent structure is never observed. The joint space of latent structure and output structure is sometimes denoted by $\mathcal{S} = \mathcal{Y} \times \mathcal{Z}$.

Full supervision

In the fully supervised setting, which is described in chapter 4, the learner is given access to a fully labeled training set $\mathcal{D}_l = \{ (x^{(j)}, y^{(j)}) : (x^{(j)}, y^{(j)}) \in \mathcal{X} \times \mathcal{Y}, j \in [1, |\mathcal{D}_l|] \}$, where each input-output pair is completely specified. The goal is to learn a mapping $\hat{y}: \mathcal{X} \to \mathcal{Y}$ from the data set $\mathcal{D}_l$, such that the mapping generalizes well to data outside the training set. Typically, a separate test set $\mathcal{D}_t = \{ (x^{(j)}, y^{(j)}) : (x^{(j)}, y^{(j)}) \in \mathcal{X} \times \mathcal{Y}, j \in [1, |\mathcal{D}_t|] \}$ is used to evaluate the induced mapping.

Provided that the training set is large enough and constitutes a sufficiently unbiased sample, this is the most powerful learning setting. However, as discussed in chapter 1, creating fully labeled training data is often expensive. We therefore turn to learning with incomplete information, with the hope that we will be able to compensate for the lack of supervision alternative sources of information.

No supervision

The most radical alternative to learning with full supervision is to learn directly from a set of raw unlabeled data $\mathcal{D}_u = \{ x^{(j)} : x^{(j)} \in \mathcal{X}, j \in [1, |\mathcal{D}_u|] \}$. In *unsupervised learning* the goal is to discover some structure of interest in $\mathcal{X}$, using the sample $\mathcal{D}_u$. The canonical types of unsupervised learning are *clustering* and *dimensionality reduction* (Ghahramani, 2004). In clustering, one seeks a partitioning of the data points in $\mathcal{X}$, such that the points are grouped in some "natural" way, for example, into high-density regions separated by regions of low density. In dimensionality reduction or *low-dimensional embedding*, one seeks to embed the data in a subspace of the original input space, such that the most important aspects of the data are retained. Another variant is *manifold embedding* (Saul and Roweis, 2003), in which one seeks to embed the data on a low-dimensional manifold rather than in a subspace. Both clustering and embedding can be used for different purposes, for example, to discover underlying structure in the data or to perform compression (Jain, 2010).

While these uses of unsupervised methods remain the most important, unsupervised learning has also been proposed as a direct replacement for supervised learning. In particular, much recent research in natural language processing has been devoted to fully unsupervised methods applied to, for example, part-of-speech tagging and syntactic dependency parsing. Since $\mathcal{X}$ itself does not provide any direct information on the mapping $\hat{y}: \mathcal{X} \to \mathcal{Y}$, if the goal is to replace supervised learning, the key information on the mapping is encoded in the model structure (Goldwater and Johnson, 2005). Although no labeled data is available during training, the learned mapping is typically evaluated against the test set $\mathcal{D}_t$, even in this scenario.

Semi-supervision

In *semi-supervised* learning, the learner has access to a small fully labeled data set $\mathcal{D}_l$, together with a large unlabeled data set $\mathcal{D}_u$, where $\mathcal{D}_l$ and $\mathcal{D}_u$ are defined as above. Typically, one assumes that the unlabeled data set is orders of magnitude larger than the labeled data set. In semi-supervised learning, the goal is to make use of the structure of $\mathcal{X}$ (discovered using $\mathcal{D}_u$) to aid in the learning of the mapping $\hat{y}: \mathcal{X} \to \mathcal{Y}$ from $\mathcal{D}_l$. The last decade has seen a steady rise of interest in research on semi-supervised learning and a multitude of novel algorithms and learning settings have been devised; see Chapelle et al. (2006) for an overview. Not surprisingly, many semi-supervised learning methods are based on a combination of supervised and unsupervised learning. These methods can be grouped into four common approaches.

The simplest approach is to apply an unsupervised learning method on $\mathcal{D}_u$, such as clustering (Freitag, 2004; Miller et al., 2004) or embedding (Turian et al., 2010; Dhillon et al., 2011), to discover informative features of the data, which can subsequently be used for standard supervised learning on $\mathcal{D}_l$.

The oldest semi-supervised approach is to let the learner leverage its own predictions in an iterative manner, a procedure known as *self-training* or *bootstrapping* (Yarowsky, 1995). In its simplest form, an initial predictor is trained on a small set of labeled data, which is then iteratively augmented with subsets of the unlabeled data, labeled by the model's own predictions. Usually a confidence-based selection strategy is used, such that only high-confidence predictions (according to some threshold) are added to the labeled data in each iteration. As discussed by Abney (2004), this approach is closely related to the expectation-maximization algorithm, which we briefly describe in section 5.2.2. This connection is also explicit in the *classification EM* algorithm of Celeux and Govaert (1992), which is derived from the perspective of combining clustering and classification. A related algorithm is the *co-training* algorithm of Blum and Mitchell (1998). In co-training, the feature space is partitioned into different *views* and separate classifiers are trained on each view, such that the classifiers are regularized to agree on unlabeled data. This algorithm was used to bootstrap a named-entity recognizer from a small set of high-precision seed rules by Collins and Singer (1999). Yet another related framework is that of *(minimum) entropy regularization*, in which the model is regularized to produce predictions of low entropy on unlabeled data (Grandvalet and Bengio, 2004; Jiao et al., 2006). We describe self-training methods in more detail in chapter 9, where we propose a simple, yet effective, self-training algorithm based on learning with ambiguous self-supervision.

Another common approach is to use the unlabeled data set to uncover the underlying geometry of the data, either directly in the form of a (truncated) graph of pairwise similarities between the data points, or by some manifold-learning technique. The geometry can then be used for learning, for example, by propagating the label information in $\mathcal{D}_l$ over the graph to the points in $\mathcal{D}_u$ (Zhou et al., 2003; Zhu et al., 2003). This approach is an instance of *transductive* learning, as it assumes access to the test set during learning. In other words, whenever additional test data arrives, the test data has to be incorporated in the geometry and the propagation has to be repeated. There are, however, inductive variants of these approaches as well, where the geometry induced from $\mathcal{D}_u$ is used as a regularizer when performing supervised learning on $\mathcal{D}_l$ (Chapelle et al., 2003; Bousquet et al., 2004; Belkin et al., 2006). Another variant is to learn the embedding and the predictor jointly using $\mathcal{D}_u \cup \mathcal{D}_l$, for example, by means of a deep neural network as proposed by Weston et al. (2008) and Collobert and Weston (2008).

Another alternative is to use a model in which the output is observed only for instances in the labeled data set, while treated as latent structure for the instances in the unlabeled data set (Nigam et al., 1999). Different loss functions are used for the labeled and unlabeled instances: fully observed instances are penalized by a loss function that takes all variables into account, while unobserved instances are penalized by a loss function that only depends indi-

rectly on the latent structure, via its influence on the observations. The model structure thus provides the link between the unlabeled and the labeled data.

Prototype supervision
In some cases, a list of prototypical members of some substructure type can be assembled without much cost. For example, while annotating the part-of-speech of each token in a corpus is costly, a set of prototypical members of each part of speech can be easily listed by a native speaker of the language. Such prototypes can be incorporated into models in different ways. For example, Haghighi and Klein (2006) use prototypes in sequence labeling tasks by aligning each word with its closest prototypes based on distributional similarity. The set of prototypes to which each word is aligned is then encoded as features in a Markov random field (MRF) — a globally normalized generative model; see Bishop (2006) — which is then trained using a standard unsupervised learning method. Another example is Velikovich et al. (2010), who use prototypical words of positive and negative polarity as constraints on the vertex distributions in graph-based semi-supervised learning to induce a polarity lexicon for sentiment analysis.

Prototypes can be defined a priori, a posteriori, or concurrently with learning. Examples of selecting prototypes a priori are the works cited above, as well as the early work of Brill and Marcus (1992). A posteriori selection of prototypes has typically been used in conjunction with word clustering. For example, Schütze (1993) and Lamar et al. (2010) first cluster all word types in a corpus and then manually label the prototypical members of each cluster with their prototypical part-of-speech tag. These prototypical labeled word types are then used to label the remaining words, based on their similarity to the prototypes.

In a sense, the bootstrapping methods of Yarowsky (1995) and Collins and Singer (1999) can also be viewed as instances of learning with prototype supervision. The small sets of high-precision rules leveraged by these authors are designed to capture the most prototypical cases, for example, that the string *New York* should always be tagged as LOCATION. While similar, the approach of Haghighi and Klein (2006) is more flexible compared to the bootstrapping approaches, in the sense that the prototypes are only used as features in an unsupervised model, so that the model is free to adapt to the data, rather than being partly constrained by fixed rules.

Indirect supervision
Sometimes we may not directly observe the structure that we seek to predict in the training set, but we may instead have access to some other signal, which carries information on the structure of interest. For example, in chapter 6, we seek to predict sentence-level sentiment, by learning form a training set in which only the document-level labels are observed. In other words, the learner has access to an indirectly labeled training set $\mathcal{D}_p = \{(x^{(j)}, y^{(j)}) : x^{(j)} \in$

$\mathcal{X}, y^{(j)} \in \mathcal{Y}(x^{(j)}), j \in [1, |\mathcal{D}_p|] \}$, where $\mathcal{Y}$ denotes the part of the output space that corresponds to the structure that is observed during training. In addition to the output space $\mathcal{Y}$, we assume a space of latent output structure $\mathcal{Z}$ and our goal is to learn the mapping $\hat{z}: \mathcal{X} \mapsto \mathcal{Z}$ from the indirectly labeled training set $\mathcal{D}_p$. In some cases, we may also want to learn the mapping $(\hat{y}, \hat{z}): \mathcal{X} \mapsto \mathcal{Y} \times \mathcal{Z}$. In case the structure in $\mathcal{Y}$ is known to always be observed at test time, our goal may instead be to learn the mapping $\hat{z}: \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{Z}$. Note that the test set $\mathcal{D}_t$ in this case contains gold standard labels, either for both y and z, or only for z, if the output y is not of interest.

If the indirectly labeled data set $\mathcal{D}_p$ is combined with a fully labeled data set $\mathcal{D}_l$, a variant of the last type of semi-supervised learning above can be performed, where the fully labeled data helps guide the learning of structure that is only latent in the indirectly labeled data. This may be effective if the indirectly labeled data set is considerably larger than the fully labeled data set, so that learning only from the latter is likely to yield a suboptimal predictor. Variants of this scenario for sentence-level sentiment analysis are explored in chapter 6.

A special case of indirect supervision, where the observed signal comes in the form of binary labels, was proposed by Chang et al. (2010). In this scenario, the learner is provided with a large data set $\mathcal{D}_b = \{ (x^{(j)}, b^{(j)} : x^{(j)} \in \mathcal{X}, y^{(j)} \in \{-1, 1\}, j \in [1, |\mathcal{D}_b|] \}$, where $b = 1$ indicates that there exists some well-formed structure $z \in \mathcal{Z}(x)$, whereas $b = -1$ indicates that none of the potential structures in $\mathcal{Z}(x)$ are well-formed. While learning is possible purely from such binary indirect supervision, the authors also consider the semi-supervised variant of this scenario, where a small set of fully labeled data $\mathcal{D}_l$ is leveraged in addition to the large set of indirectly labeled data. It is easy to show that the loss function proposed by Chang et al. (2010) is a special case of the latent hinge loss function (see section 5.2.1) applied to indirectly labeled data, where the observations are restricted to be binary variables.

Ambiguous supervision

In the indirectly supervised scenario, there is a separation between the observed and the latent parts of the inference space, such that one part is always fully observed and the other part is always fully unobserved. A related scenario is learning with *ambiguous supervision*, in which an arbitrary part of the output may be only ambiguously specified. The learner thus is provided with a ambiguously labeled training set $\mathcal{D}_a = \{ (x^{(j)}, \boldsymbol{y}^{(j)}) : x^{(j)} \in \mathcal{X}, \boldsymbol{y}^{(j)} \subseteq \mathcal{Y}(x^{(j)}), j \in [1, |\mathcal{D}_a|] \}$, where the output $\boldsymbol{y} \subseteq \mathcal{Y}(x)$ is an ambiguous labeling of the input x. Note that if $\boldsymbol{y} = \mathcal{Y}(x)$, the training set contains no constraints on the output space, so that this scenario reduces to standard unsupervised learning. From the perspective of inference and learning, there is no fundamental difference between ambiguous and indirect supervision, as discussed in more detail in section 5.2.

In this dissertation, we leverage ambiguous supervision in two ways. First, in chapter 9, we introduce a self-training algorithm that uses automatically inferred ambiguous supervision, with application to syntactic dependency parsing. Second, in chapter 10, we encode semi-automatically derived constraints on both tokens and types as ambiguous supervision for cross-lingual part-of-speech tagging.

Higher-level constraints

In the above scenarios, supervision comes in the form of fully or partially observed outputs. As we shall see in section 5.2.1, learning with such observations corresponds to guiding the learner away from *unconstrained* hypotheses towards *constrained* hypotheses. In the fully supervised scenario, there is only one unique hypothesis towards which to move, whereas in the ambiguously supervised scenario, the learner is guided towards hypotheses that are consistent with the ambiguous labeling $\boldsymbol{y} \subseteq \mathcal{Y}(x)$ — we refer to $\boldsymbol{y}$ as the *observation inference space* — at the expense of hypotheses in the unconstrained inference space $\mathcal{Y}(x)$. More generally, any (ambiguously) labeled substructure corresponds to a (set of) hard constraint(s) on the observation inference space, such that the constraints decompose with the model structure.

However, just as many useful cost functions are non-decomposable (see section 4.1.2), many constraints that are potentially useful for learning cannot be expressed in terms of full or ambiguous supervision that decompose with the model structure in this way. For example, the constraint that a part-of-speech tag sequence should contain at least one verb corresponds to a factor that involves every position in the output sequence. Such higher-order constraints may be particularly useful for guiding learning in the presence of latent variables. Unfortunately, due to their non-local nature, higher-order constraints are more difficult to incorporate into learning and inference.

Nevertheless, there has been a flurry of work on methods for learning with higher-level constraints. In almost all work, the constraints are assumed to be linear in some non-decomposable model features. When combined with a linear score function, this means that one can always revert to inference based on integer linear programming (ILP). Since this is costly, other methods are often preferred, such as beam-search (Chang et al., 2007), variational inference (Bellare et al., 2009; Liang et al., 2009; Ganchev et al., 2010), or dual decomposition (Das et al., 2012). ILP-based inference was primarily used in early work on learning sequence-models with structural constraints (Roth and tau Yih, 2004; Punyakanok et al., 2004, 2005; Toutanova et al., 2005).

A more unified approach to learning with inference constraints was proposed by Chang et al. (2008, 2012), under the term *constrained-conditional models* (CCM). The *constraint-driven learning* (CODL) framework, proposed by Chang et al. (2007), is a semi-supervised variant of CCM, which allows combining inference constraints with unlabeled data. A variant of constraint-driven learning which handles constrained latent structure was later intro-

duced by Chang et al. (2010). Similar methods have also been proposed by Mann and McCallum (2008) and Ganchev et al. (2010), under the names *generalized expectation criteria* and *posterior regularization*, respectively. As shown by Ganchev et al. (2010), all these methods correspond to different approximations to the fully Bayesian approach of Liang et al. (2009). Additional connections between posterior regularization and constraint-driven learning are made by Samdani et al. (2012), who show that these are specific cases of a general algorithm based on variational EM, corresponding to different settings of an entropy-controlling temperature parameter.

Finally, Hall et al. (2011) propose another related framework, which permits general reranking loss functions to be applied to k-best lists in a self-training algorithm. This enables learning with multiple loss functions, for example, for adapting a syntactic parser specifically to a machine translation task (Katz-Brown et al., 2011). Since this approach is based on reranking hypotheses in a k-best list, the loss function can be based on arbitrary constraints, with the caveat that one may need to set k arbitrarily large when learning with loss functions that correspond to hard constraints, since at least one hypothesis that obeys the constraints needs to be found among the k best hypotheses.

5.2 Structured Latent Variable Models

Structured latent variable models constitute a key modeling tool in subsequent chapters. The key difference from the models described in chapters 3 and 4 is that in addition to a space of inputs $\mathcal{X}$ and a space of outputs $\mathcal{Y}$, we assume a space of latent structure $\mathcal{Z}$. The distinction between these spaces is that the learner is assumed to always observe $\mathcal{X}$, while $\mathcal{Y}$ is only assumed to be observed during training and $\mathcal{Z}$ is assumed to never be directly observed.

In unsupervised learning, the model is thus specified by means of $\mathcal{X}$ and $\mathcal{Z}$ alone and the model structure, that is, the relationship between $\mathcal{X}$ and $\mathcal{Z}$, is fundamental to successful learning. Since no outputs are observed, only generative models can be used for unsupervised learning.[1] For example, in one of the simplest structured latent variable models, the hidden Markov model (HMM) for sequence labeling, $\mathcal{X}$ is the space of input sequences and $\mathcal{Z}$ is the space of label sequences. By assuming that each input sequence is generated by a latent label sequence and by maximizing the likelihood of the observed sequences in the training set $\mathcal{D}_u$, according to this model of generation, a mapping $\mathcal{X} \to \mathcal{Z}$ can be induced by application of Bayesian inversion. Similarly, when $\mathcal{Y}$ only indirectly or ambiguously specifies what one seeks to predict, an additional space of latent structure $\mathcal{Z}$ can be included in the model.

[1] While out of scope for the present discussion, we note that there are alternative ways to train unsupervised models, such as *contrastive estimation* (Smith and Eisner, 2005a,b)

In this case, both generative and discriminative models can be used; in a generative model, the input is assumed to be generated by $\mathcal{Y}$ and $\mathcal{Z}$ jointly, while in a discriminative model, the output $\mathcal{Y}$ is typically modeled as conditioned on $\mathcal{X}$ and $\mathcal{Z}$, or on $\mathcal{X}$ via $\mathcal{Z}$.

Thus, latent variable models are highly flexible and, as we show in subsequent chapters, these models can, despite their simplicity, be used to obtain state-of-the-art results for several different tasks in natural language processing. In summary, we apply latent variable models to the following tasks:

- In chapter 6, such models are used for learning to predict sentence-level sentiment from indirect supervision in which only the document-level sentiment is observed.
- In chapter 8, such models are used for inducing (cross-lingual) word clusters for use in semi-supervised (cross-lingual) named-entity recognition and syntactic dependency parsing.
- In chapter 9, such models are used for a novel self-training algorithm, which is applied to target language adaptation in cross-lingual syntactic dependency parsing.
- In chapter 10, such models are used to learn ambiguously supervised cross-lingual part-of-speech taggers.

5.2.1 Latent Loss Functions

The structured prediction framework from chapter 3 can readily incorporate latent structure. However, in order to achieve this, we need to redefine the score function slightly to score the input as well as the output and latent structure jointly. Thus, rather than the score function $s: \mathcal{X} \times \mathcal{Y} \times \Theta \rightarrow \mathbb{R}$, which was assumed in previous chapters, we instead use a score function $s: \mathcal{X} \times \mathcal{Y} \times \mathcal{Z} \times \Theta \rightarrow \mathbb{R}$, where $\mathcal{Z}(x)$ denotes the space of latent structures spanning input x. The space of latent structures is defined in terms of substructure indicator vectors, just like the space of (observed) output structure $\mathcal{Y}$, so that $\mathcal{Y}(x) \times \mathcal{Z}(x) \subset \{0,1\}^{\mathcal{J}(x)}$ for some index set $\mathcal{J} = \mathcal{J}_{\mathcal{Y}} \cup \mathcal{J}_{\mathcal{Z}}$. The only difference between $y \in \mathcal{Y}$ and $z \in \mathcal{Z}$ is that y is assumed to be observed during training, while z is assumed to be unobserved. With ambiguous supervision, the learner observes partially constrained outputs $\boldsymbol{y} \subseteq \mathcal{Y}(x)$, so that the space of latent structure correspond to all hypotheses that are consistent with these constraints.

While the representation remains the same, in order to learn in this setting, the surrogate loss functions from section 4.1.3 need to be modified to incorporate latent structure. Since the loss functions are defined in terms of the predictions of the model, we need to first consider how latent structure may be incorporated in prediction. Essentially, this can be achieved in two ways. First, in *maximum a posteriori* (MAP) inference, a maximization is performed

over both the output space and the latent space:

$$(\hat{y}, \hat{z})(x) = \operatorname*{arg\,max}_{(y', z') \in \mathcal{Y}(x) \times \mathcal{Z}(x)} s_\theta(x, y', z') . \tag{5.1}$$

Second, in *marginal MAP* inference, a maximization is performed over the output, while simultaneously marginalizing over the space of latent structure:

$$\hat{y}(x) = \operatorname*{arg\,max}_{y \in \mathcal{Y}(x)} \sum_{z' \in \mathcal{Z}(x)} \exp\left\{ s_\theta(x, y, z') \right\} .$$

Unfortunately, while marginalizing over the latent structure for a fixed input $y \in \mathcal{Y}(x)$ can often be performed efficiently, exact joint maximization and marginalization is often computationally intractable.[2] This is, however, not a concern for the models studied in this dissertation, since we are either interested in both the latent structure and the output structure, or we are assuming that the output is observed and we are seeking to predict the optimal latent structure for this fixed output. The latter occurs in the case of sentence-level sentiment analysis (see chapter 6), where document-level sentiment may sometimes be observed at test time. In that case we seek to predict the MAP assignment of latent structure, for the fixed observed document label. In case the document label is not observed, we typically want to predict it as well, although we could also consider marginalizing over its possible assignments.

We now turn to the latent variants of the surrogate loss functions, assuming that eq. (5.1) will be used for prediction at test time. The perceptron loss in eq. (4.7) can be naturally generalized to incorporate latent variables as follows. First, the observed output y is replaced by a MAP assignment of the latent structure $\hat{z}(x, y) \in \mathcal{Z}(x)$, keeping the input x and the observed output y fixed:

$$\hat{z}(x, y) = \operatorname*{arg\,max}_{z' \in \mathcal{Z}(x)} s_\theta(x, y, z') .$$

Second, the prediction $\hat{y}(x) \in \mathcal{Y}(x)$ is replaced by the joint MAP assignment in eq. (5.1). Combining these, we obtain the *latent perceptron loss*

$$L_{\text{lat-per}}(x, y, \theta) = -\left(\hat{s}_\theta(x, y) - \hat{s}_\theta(x)\right),$$

where

$$\hat{s}_\theta(x, y) = s_\theta\left(x, y, \hat{z}(x, y)\right)$$

and

$$\hat{s}_\theta(x) = s_\theta\left(x, \hat{y}(x), \hat{z}(x)\right).$$

Note that $\hat{s}_\theta(x)$ overloads the notation from chapters 3 and 4.

[2] Efficient approximate algorithms for marginal MAP inference were recently proposed by Jiang et al. (2011) and Liu and Ihler (2011).

As with the log loss in eq. (4.8), we can replace the maximization in the latent perceptron loss with the soft-max, which results in the *latent log loss*

$$L_{\text{lat-log}}(x, y, \theta) \;=\; -\left(\log Z_\theta(x, y) - \log Z_\theta(x)\right), \tag{5.2}$$

where

$$Z_\theta(x, y) \;=\; \sum_{z' \in \mathcal{Z}(x)} \exp\left\{s_\theta(x, y, z')\right\}$$

is the partition function with x and y fixed, and

$$Z_\theta(x) \;=\; \sum_{(y', z') \in \mathcal{Y}(x) \times \mathcal{Z}(x)} \exp\left\{s_\theta(x, y', z')\right\}$$

is the partition function with only x fixed. Performing structural risk minimization with the latent log loss corresponds to maximum likelihood estimation — or maximum a posteriori estimation when a regularizer corresponding to a prior distribution is used — in a probabilistic model with latent variables. When the model structure factorizes according to an undirected graph, such a model is often referred to as a hidden conditional random field (HCRF; Quattoni et al., 2007).[3]

Analogously, the hinge loss in eq. (4.9) can be extended with latent variables to give the margin-scaled *latent hinge loss*

$$L_{\text{lat-hinge}}(x, y, \theta) \;=\; -\left(\hat{s}_\theta(x, y) - \hat{s}_\theta(x, y, C)\right),$$

where

$$\hat{s}_\theta(x, y, C) \;=\; \max_{(y', z') \in \mathcal{Y}(x) \times \mathcal{Z}(x)} \left[s_\theta(x, y', z') + C(y, y')\right].$$

This loss corresponds to the margin-scaled variant of the *latent structural support vector machine* (LSSVM; Yu and Joachims, 2009).

Recently, Pletscher et al. (2010) and Hazan and Urtasun (2010) generalized both the latent hinge loss and the latent log loss, by introducing a temperature term which controls the entropy of the conditional distribution $p_\theta(y, z \mid x)$. Applying the latent log loss to this distribution results in a family of models, of which the LSSVM and the HCRF are special cases which correspond to different values of the temperature term. Similar entropy regularization variants were proposed by Samdani et al. (2012) in the context of constrained models trained with expectation-maximization and by Tu and Honavar (2012) in the context of constrained models for unsupervised syntactic dependency parsing.

[3] In the natural language processing community, the term *HCRF* is often associated with a particular model structure. However, in this dissertation we use the term more broadly to refer to any structured model trained with the latent log loss.

5.2.2 Learning with Latent Variables

In the remainder of this dissertation, the only latent loss function that we consider is the latent log loss. This choice do not have a substantial impact on the generality of our results. As with the standard log loss, we use gradient-based optimization to minimize the resulting regularized empirical risk. Unfortunately, while the log loss is convex, this is not true of the latent log loss, due to the marginalization over the latent structure. The same holds for all loss functions that include latent structure. Therefore, any gradient-based optimization method is only guaranteed to find a local minimum of the objective function. In practice, this issue can be partly remedied by running the optimization multiple times with different initial parameter vectors and selecting the solution with the minimal regularized empirical risk. In some cases, prior knowledge may be used to find an initial parameter vector that is closer to the global optimum. Below, we provide the gradients of the latent log loss in the globally normalized discriminative case, as well as in the locally normalized generative case.

The globally normalized discriminative case

The gradient of the latent log loss has a similar form as that of the standard log loss in eq. (4.12). However, whereas the gradient of the log loss is expressed as the difference between observed and expected feature counts, the gradient of the latent log loss involves a difference between two expectations. The gradient of the log-partition function $\log Z_\theta(x, y)$, where the output y is fixed to its observed value, is given by the expected feature counts, conditioning on x and y:

$$\frac{\partial}{\partial \theta} \log Z_\theta(x, y) = \mathop{\mathbb{E}}_{p_\theta(z'|x,y)} \left[\Phi(x, y, z')\right].$$

Similarly, the gradient of the log-partition function $\log Z_\theta(x)$ is given by the expected feature counts, conditioning only on x:

$$\frac{\partial}{\partial \theta} \log Z_\theta(x) = \mathop{\mathbb{E}}_{p_\theta(y',z'|x)} \left[\Phi(x, y', z')\right].$$

Putting these together, we obtain the gradient of the latent log loss

$$\frac{\partial}{\partial \theta} L^{\text{latent}}_{\log}(x, y, \theta) = -\left(\mathop{\mathbb{E}}_{p_\theta(z'|x,y)} \left[\Phi(x, y, z')\right] - \mathop{\mathbb{E}}_{p_\theta(y',z'|x)} \left[\Phi(x, y', z')\right] \right). \tag{5.3}$$

Note that although the latent structure is marginalized out in the latent log loss, for reasons discussed above, after training the model, MAP inference is typically used for prediction.

The locally normalized generative case

In the supervised case (see section 4.1.3), when applied to a locally normalized generative log-linear model (see section 3.4.2), the negative log-likelihood decomposes into a sum of log losses over local factors. However, the marginalization over latent variables complicates matters, since it prevents us from directly decomposing the loss over individual factors. For ease of exposition, let us ignore the observed outputs $y \in \mathcal{Y}$ and assume that we only have observations $x \in \mathcal{X}$ and latent structure $z \in \mathcal{Z}$. The contribution of each instance x to the negative log-likelihood is

$$-\log p_\beta(x) \;=\; -\log \sum_{z \in \mathcal{Z}(x)} p_\beta(x, z)\,.$$

There are essentially two standard ways to minimize this loss function. The most common approach is to use the expectation-maximization (EM) algorithm (Dempster et al., 1977), which alternates between the following two steps until convergence (note that minimizing the negative marginal log loss is equivalent to maximizing the marginal log loss):

Expectation Fix the current parameters $\beta^{\text{old}} \leftarrow \beta$ and infer the conditional distribution over the hidden variables: $p_{\beta^{\text{old}}}(z \mid x) = \frac{p_{\beta^{\text{old}}}(x,z)}{p_{\beta^{\text{old}}}(x)}$.

Maximization Find the new parameters β that maximize the expected joint likelihood: $\beta \leftarrow \arg\max_{\beta'} \mathbb{E}_{p_{\beta^{\text{old}}}(z|x)}\left[p_{\beta'}(x, z)\right]$.

The parameters β are either initialized to some random value or based on prior knowledge. It can be shown that the EM procedure corresponds to coordinate ascent on iteratively refined lower bounds of the marginal likelihood $p_\beta(x)$, and that this procedure is guaranteed to converge to a local maximum of the marginal likelihood (Neal and Hinton, 1999). As in the supervised case, when the local factors are categorical distributions, the maximization step has a simple close-form solution, whereas with a log-linear parameterization, gradient-based optimization methods are typically employed in the maximization step (Chen, 2003; Berg-Kirkpatrick et al., 2010).

Another approach, proposed by Salakhutdinov et al. (2003), is to apply gradient-based optimization to directly minimize the negative marginal log-likelihood. As observed by Salakhutdinov et al., the gradient of the marginal log-likelihood can be expressed as the expectation of the gradient of the joint log-likelihood with respect to the posterior distribution over the latent structure. This can be easily shown by application of the chain rule to the gradient of the negative marginal log-likelihood. Recall that for any differentiable function $f : v \rightarrow \mathbb{R}$ it holds that

$$\frac{\partial}{\partial v} \log f(v) \;=\; \frac{1}{f(v)} \frac{\partial}{\partial v} f(v) \qquad \Leftrightarrow \qquad \frac{\partial}{\partial v} f(v) = f(v) \frac{\partial}{\partial v} \log f(v)\,.$$

Consequently, we have that

$$\begin{aligned}
\frac{\partial}{\partial \beta} - \log p_\beta(x) &= -\frac{1}{p_\beta(x)} \frac{\partial}{\partial \beta} p_\beta(x) \\
&= -\frac{1}{p_\beta(x)} \sum_{z' \in \mathcal{Z}(x)} \frac{\partial}{\partial \beta} p_\beta(x, z') \\
&= -\frac{1}{p_\beta(x)} \sum_{z' \in \mathcal{Z}(x)} p_\beta(x, z') \frac{\partial}{\partial \beta} \log p_\beta(x, z') \\
&= -\sum_{z' \in \mathcal{Z}(x)} p_\beta(z' \mid x) \frac{\partial}{\partial \beta} \log p_\beta(x, z') \\
&= \mathop{\mathbb{E}}_{p_\beta(z' \mid x)} \left[\frac{\partial}{\partial \beta} - \log p_\beta(x, z') \right].
\end{aligned}$$

By plugging in the gradient of the logarithm of the joint distribution, which decomposes into gradients over local log losses (see section 4.1.3), this gradient is easily computed.

The direct gradient approach was popularized in the natural language processing community by Berg-Kirkpatrick et al. (2010), who observed considerably better results with this method compared to EM in a number of unsupervised learning tasks. However, Li et al. (2012) report better results with EM for a similar task, so the relative merit of these methods is still a topic of debate. In this dissertation, we restrict ourselves to the direct gradient approach when learning locally normalized log-linear models in chapter 10, as we observed better results with this approach compared to EM in a preliminary study. We note that Salakhutdinov et al. (2003) propose a procedure for combining these methods, which could potentially lead to better results.

6. Sentence-Level Sentiment Analysis with Indirect Supervision

In the previous chapter, we described methods for learning with incomplete supervision and we argued for the versatility of structured discriminative latent variable models to this end. In this chapter, we study an application of such models to the task of fine-grained sentiment analysis, a central task in the field of opinion mining and sentiment summarization; see chapter 2. Specifically, we propose to jointly model sentence- and document-level sentiment, treating the former as latent structure and the latter as indirect supervision. Typical supervised learning approaches to sentence-level sentiment analysis rely on sentence-level supervision. While such fine-grained supervision rarely exists naturally and thus requires labor intensive manual annotation effort (Wiebe et al., 2005), indirect supervision is naturally abundant in the form of online product-review ratings. Thus, learning to analyze fine-grained sentiment strictly from indirect supervision, would allow us to sidestep laborious annotation effort.

To provide some intuitions about what such a model ought to look like, we first describe the assemblage of a large data set of product reviews, where a small subset of the reviews are manually annotated with sentence-level sentiment. This data set is used to empirically evaluate the various models introduced in this chapter and has been made publicly available. After proposing a set of natural baselines in this setting, based on polarity lexica as well as machine learning, we introduce our indirectly supervised structured latent variable model. Empirical results suggest that sentence-level sentiment labels can indeed be learned solely from document-level supervision to some degree — the structured latent variable model outperforms all baselines — but that such indirect supervision may be too weak. Based on the observed shortcomings of the indirectly supervised model, we propose two semi-supervised extensions, where the large amount of document-level supervision is complemented by a small amount of sentence-level supervision. The semi-supervised models are shown to rectify the most pressing shortcomings of the indirectly supervised model. The chapter is concluded with a discussion of and comparison with some recent related work.

Before moving on, note that this chapter only considers the monolingual (English) setting, while the remainder of the dissertation is devoted to cross-lingual learning methods (using similar structured latent variable models). Although we do not study sentiment analysis in the cross-lingual setting, the approaches developed in subsequent chapters may certainly be useful for cross-lingual sentiment analysis as well.

Table 6.1. *Number of sentences per document sentiment category for each domain in a large training sample. There are* 9572 *documents for each pair of domain and document sentiment, for a total of* 143 580 *documents.*

	POS	NEG	NEU	Total
Books	56 996	61 099	59 387	177 482
Dvds	121 740	102 207	131 089	355 036
Electronics	73 246	69 149	84 264	226 659
Music	65 565	55 229	72 430	193 224
Videogames	163 187	125 422	175 405	464 014
Total	480 734	430 307	522 575	1 416 415

6.1 A Sentence-Level Sentiment Data Set

There are several freely available data sets annotated with sentiment at various levels of granularity; comprehensive lists of references can be found in Pang and Lee (2008) and Liu (2010). For training the models developed in this chapter, a data set which is annotated both at the sentence-level and at the document-level is required. The data set that was used in the empirical study of McDonald et al. (2007) is close in spirit, but unfortunately it lacks neutral documents. Since neutral reviews are abundant in most domains, this is an unrealistic over-simplification. We therefore set out to collect a large corpus of consumer reviews from a range of domains, where each review is annotated with document-level sentiment automatically extracted from its star rating. For evaluation purposes, we further set out to assemble a small set of reviews, in which each review is additionally manually annotated at the sentence level.

A training set was created by sampling a total of 150 000 positive, negative and neutral product reviews from five different domains: *books*, *dvds*, *electronics*, *music* and *videogames*. The reviews were labeled with document-level sentiment, using the following scheme: reviews with 1 or 2 stars were labeled as *negative* (NEG), reviews with 3 stars were labeled as *neutral* (NEU) and reviews with 4 and 5 stars were labeled as *positive* (POS). After removing duplicate reviews, a balanced set of 143 580 reviews remained. Of course, sampling reviews in this way is a simplification, as naturally occurring product reviews are typically not balanced with respect to rating. Each review was split into sentences using standard heuristics. As can be seen from the detailed sentence level statistics in table 6.1, the total number of sentences is roughly 1.5 million. Note that this training set only has labels at the document level, as reviewers do not annotate more fine-grained sentiment in consumer reviews.

The same procedure was used to create a smaller separate test set consisting of 300 reviews, again uniformly sampled with respect to the domains and document sentiment categories. After removing duplicates, 97 positive, 98

Table 6.2. *Number of documents per sentiment category (left) and number of sentences per sentence-sentiment category (right) in the labeled test set, across domains.*

	Documents per category				Sentences per category			
	POS	NEG	NEU	Total	POS	NEG	NEU	Total
Books	19	20	20	59	160	195	384	739
Dvds	19	20	20	59	164	264	371	799
Electronics	19	19	19	57	161	240	227	628
Music	20	20	19	59	183	179	276	638
Videogames	20	20	20	60	255	442	335	1032
Total	97	99	98	294	923	1320	1593	3836

neutral and 99 negative reviews remained. Two annotators were assigned to annotate each sentence in the test set with the following categories: *positive* (POS), *negative* (NEG), *neutral* (NEU), *mixed* (MIX), and *not related* (N/R).[1]

Annotation guidelines

The following annotation guidelines were provided for the annotators:

1. The categories POS and NEG are to be assigned to sentences that clearly express positive and negative sentiment, respectively.
2. The category NEU is to be assigned to sentences that express sentiment, but are neither clearly positive nor clearly negative, such as *The image quality is not good, but not bad either.*
3. The category MIX is to be assigned to sentences that express both positive and negative sentiment, such as *The plot is great, but the acting sucks!.*
4. The N/R category is to be assigned to sentences that contain no sentiment, as well as to sentences that express sentiment about something other than the target of the review.
5. All but the N/R category are to be assigned to sentences that either express sentiment by themselves, or that are part of an expression of sentiment spanning several sentences.

Regarding item 3, it would of course be preferable to perform an even more fine-grained analysis, where each chunk of text is assigned a unique sentiment. The guideline in item 5 allows us to annotate, for example, *Is this good? No* as negative, despite the fact that this expression is split into two sentences in the preprocessing step. While this approach lets us circumvent such problematic cases, it does suggest that sentence-level sentiment may not always be an adequate level of analysis.

[1] One half of the reviews were annotated by the author and the other half by Ryan McDonald, with an overlap of 175 sentences in order to evaluate inter-annotator agreement, as further discussed below.

Table 6.3. *Distribution of sentence-level sentiment labels (columns) by document-level label (rows). Each cell shows the percentage of sentences in the test set that fall into the corresponding category.*

	Sentence label		
	POS (%)	NEG (%)	NEU (%)
POS	53	8	39
NEG	5	62	33
NEU	14	35	51

Test set statistics

The total number of annotated sentences in the test set is close to 4000. In order to simplify our experiments and make our evaluation more robust, we subsequently merge the MIX and N/R categories into the NEU category; that is, NEU can be considered a type of "catch-all" category. Statistics of the test set as per domain are found in table 6.2, while table 6.3 shows the distribution of sentence-level sentiment for each document sentiment category. From table 6.3, it is evident that the sentence-level sentiment is aligned with the document-level sentiment. However, reviews from all categories contain a substantial fraction of neutral sentences and a non-negligible fraction of both positive and negative sentences.

Inter-annotator agreement

In order to estimate the inter-annotator agreement of the test set, where the sentences in each document were annotated by a single annotator, we measured agreement on a separate set of 175 sentences, where each sentence was annotated by both annotators. The overall raw agreement was 86%, with a Cohen's Kappa value (Cohen, 1960) of $\kappa = 0.79$. The class-specific agreements, measured in terms of F_1 score treating the labels from one annotator as the gold standard, were respectively 83%, 93% and 82%, for the POS, NEG and NEU category. The agreement is comparable to previously reported agreement in this context. For example, Wilson et al. (2005) report a raw agreement of 82% ($\kappa = 0.72$) when annotating utterances with positive, negative and neutral sentiment.[2] The annotated test set is publicly available from https://github.com/oscartackstrom/sentence-sentiment-data.

6.2 Baseline Models

Before introducing the structured latent variable models, let us briefly consider a set of natural baselines in this setting. In addition to a baseline based

[2]Note that they assume that utterances are pre-classified as being subjective or objective.

Table 6.4. *Number of entries for each rating in the MPQA polarity lexicon.*

Rating	Number of entries
−1.0	3614
−0.5	1286
0.0	591
0.5	1001
1.0	1717
Total	8209

on rule-based matching against a polarity lexicon, we consider two similar baselines based on supervised machine learning.

A polarity-lexicon baseline

As discussed in section 2.5, polarity lexica are commonly used for sentiment analysis at different levels of granularity. As a first experiment, we examine the potential use of the Multi-Perspective Question Answering (MPQA) Subjectivity Lexicon (Wilson et al., 2005) for sentence-level sentiment prediction.[3] The MPQA lexicon is a polarity lexicon that rates a list of words (and phrases) on a scale in the interval $[-1.0, 1.0]$, where values less than zero is assigned to words that are assessed to convey negative sentiment and values above zero are assigned to words that are assessed to convey positive sentiment. It was first used by Wilson et al. (2005) for expression-level sentiment classification. The distribution of ratings in the lexicon is shown in table 6.4.

Our first baseline *VoteFlip*, uses the MPQA lexicon as a source of word polarities. In order to classify a sentence, each token in the sentence is first matched against the lexicon. These matches, along with their corresponding polarities, are then fed into the *vote-flip* algorithm (Choi and Cardie, 2009), a simple rule-based algorithm shown in algorithm 3. In essence, the polarity of the sentence is predicted based on the number of positive and negative lexicon matches, taking negations into account with a simple heuristic which flips the polarity in case the sentence contains an odd number of negation words.

As we shall see in section 6.4, applying the *VoteFlip* heuristic to the manually annotated test set results in fairly low classification and retrieval performance. This is not surprising, for several reasons. First, the lexicon is not exhaustive, which means that many potential matches are missed. Second, sentences such as *It would have been good if it had better guitar* will be misclassified as neither context, nor syntactic/semantic structure are taken into account. Third, a sentence may be positive or negative even when no

[3] Although lexica with a broader coverage can be found in the literature (Mohammad et al., 2009; Velikovich et al., 2010), we use the MPQA lexicon, since it is publicly available. The lexicon can be obtained from http://mpqa.cs.pitt.edu/ — February 19, 2013.

Algorithm 3 Vote-Flip Heuristic

```
x: Input sentence (x_i ∈ X_s)
P: Polarity lexicon (P: V → [−1, 1])
N: Set of negation words (N ⊂ V)
procedure VoteFlip(x, P, N)
    pos ← CountPositive(x_i, P)
    neg ← CountNegative(x_i, P)
    flip ← HasOddNumberNegations(x_i, N)
    if (pos > neg ∧ ¬flip) ∨ (pos < neg ∧ flip) then
        return pos
    else if (pos > neg ∧ flip) ∨ (pos < neg ∧ ¬flip) then
        return neg
    else
        return neu
    end if
end procedure
```

individual word is considered to be polarized. For example, *The book carries you from the first steps to far into the language* is positive, although no single phrase can be considered to be unambiguously positive in isolation. Fourth, the same sentence may be positive in some contexts while negative in others and this can not be captured by a generic polarity lexicon. For example, the sentence *It made me fall asleep* expresses a negative sentiment when used to describe a book or movie, but may express positive sentiment when used to describe relaxation music. Sentences such as these need to be classified either based on our world knowledge or with respect to the context in which they occurs.

Furthermore, although a negation detector (Councill et al., 2010) may address some scope problems, such as that in *Nothing negative about it*, there are numerous other scoping issues that cause errors, such as identifying the scope of the modal/hypothetical in *It would have been good if it had better guitar*. Even with an exhaustive lexicon and perfect knowledge of negation scope, simply flipping the polarity of the negated words is often not an adequate strategy. Consider, for example, *Well, I guess it's not terrible*, where flipping the negative polarity of *terrible* yields an incorrect result. Phrase-polarity lexica (Velikovich et al., 2010), could potentially handle some of these problems.

Two machine-learning baselines

These considerations have led to the use of machine learning for classifying fine-grained sentiment in context (Wilson et al., 2005; Choi and Cardie, 2009; Blair-Goldensohn et al., 2008; Velikovich et al., 2010). In these approaches, the sentence-level sentiment is learned using features from a lexicon in con-

junction with both syntactic and context features. The downside is that these approaches rely on the existence of a corpus of text labeled with sentiment at the sentence level.

In this work, we instead aim to develop a model that can circumvent this restriction and learn to predict sentence-level sentiment from product reviews which have been labeled only at the document level. Before describing our proposed approach, we describe two simple machine-learning baselines for learning with this level of supervision. The first baseline, which we term *sentence-as-document* (SentAsDoc), splits the training documents into sentences and assigns each sentence the label of the document from which it was extracted. This new training set is then used to train an unstructured log-linear classifier. Because documents often contain sentences with sentiment that differs from the overall document sentiment (see table 6.3), this is a rather crude approximation. The second baseline, *document-as-sentence* (DocAsSent), trains a log-linear document classifier on the training data in its natural form. This baseline can be seen as either treating training documents as long sentences — hence the name — or as treating test sentences as short documents. Details of the features used to train the baseline models are given in section 6.4. Results for the baselines are given in table 6.5, together with the results the structured latent variable models described in the next section. While the machine learning baselines improves on the lexicon-based VoteFlip baseline, both DocAsSent and SentAsDoc are based on the unrealistic assumption that the observed document label is a good proxy for all of the sentences in the document; this shortcoming is likely to degrade prediction accuracy.

6.3 A Discriminative Latent Variable Model

The distribution of sentence-level sentiment in the annotated data (table 6.3) suggests that reviews typically do contain sentences from one dominant class. However, reviews from each group also contain a substantial fraction of sentences that express a sentiment divergent from the dominant class. Specifically, positive reviews primarily consist of positive sentences, as well as a significant number of neutral sentences and a small number of negative sentences, while the opposite holds for negative reviews. Neutral documents, on the other hand, are dominated by neutral sentences, followed by a little over 30% negative sentences and approximately 15% positive sentences. When combined with the problems raised in the previous section, this observation suggests a model where sentence level classifications are correlated with the observed document label, but have the flexibility to disagree when evidence from the sentence or local context suggests otherwise.

In order to devise such a model, we start with the supervised *fine-to-coarse* sentiment model described by McDonald et al. (2007). Recapitulating from

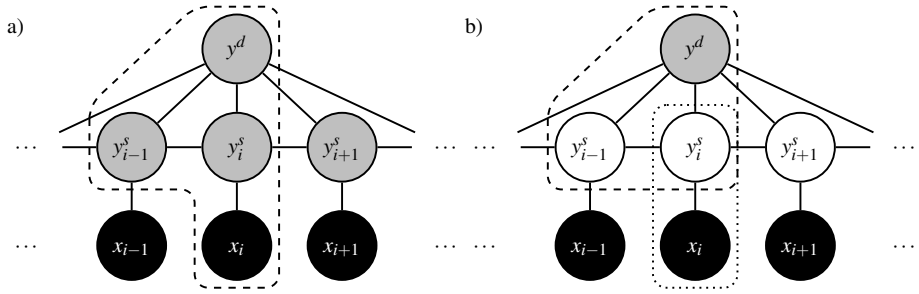

Figure 6.1. a) Outline of graphical model from McDonald et al. (2007). b) Identical model with latent sentence level states. Grey nodes indicate variables observed during training and light nodes indicate latent variables. The black sentence nodes are always fixed and observed. Dashed and dotted regions indicate the two maximal cliques at position i. Note that the document and input nodes belong to different maximal cliques in the right model, while they belong to the same maximal clique in the left model.

section 3.2, let $x = (x_1, x_2, \ldots, x_{|x|}) \in \mathcal{X}$ be a document composed of a sequence of sentences $x_i \in \mathcal{X}_s$ and let $y = (y^d, y^s) \in \mathcal{Y}(x)$, where $y^d \in \mathcal{Y}_d(x)$ denotes the document-level sentiment variable, so that $\mathcal{X} = \mathcal{X}_s^{|x|}$, and $y^s = y^s_1 y^s_2 \ldots y^s_{|x|} \in \mathcal{Y}_s(x)$ denotes the sequence of sentence-level sentiment variables. As before, we assume a linear score function which scores all variables jointly (see chapter 3):

$$s_\theta(x, y) = \theta^\top \Phi(x, y) = \theta^\top \Phi(x, y^d, y^s)\,.$$

Based on the annotation scheme described above, y^d and each y^s_i all take values in the set $\mathcal{S} = \{\text{POS}, \text{NEG}, \text{NEU}\}$. Sometimes, we will write $s_\theta(x, y^d, y^s)$ in place of $s_\theta(x, y)$ to highlight particular document- or sentence-level assignments. We hypothesize that there is a sequential relationship over sentence-level sentiment and that the document-level sentiment is influenced by all sentence-level sentiment (and vice versa). Figure 6.1a shows the undirected graphical model (Koller and Friedman, 2009) reflecting this idea. A first order Markov assumption is made, according to which each sentence variable y^s_i is independent of all other variables, conditioned on the input x_i, the document variable y^d and its adjacent sentences, y^s_{i-1}/y^s_{i+1}. This corresponds to the following factorization of the feature function:

$$\Phi(x, y) = \sum_{i=1}^{|x|} \phi(x, y^d, y^s_i, y^s_{i-1})\,.$$

With this factorization, the problem is reduced to a sequence-labeling problem (see section 3.2), by viewing (y^d, y^s_i) as a complex label and restricting sentence-level transitions to keep the assignment of y^d fixed (McDonald et al., 2007). The strength of this model is that it allows sentence- and document-

level classifications to influence each other while giving them freedom to disagree when influenced by the input. McDonald et al. showed that this model can increase the accuracy of the predictions at both levels. Unfortunately, it requires labeled data at both levels of analysis for training.

We are interested in the common case where document labels are available, for example, from star-rated consumer reviews, but sentence labels are not. A modification to the model from fig. 6.1a is to treat all the sentence sentiment variables as latent variables, as shown in fig. 6.1b. When the underlying model from fig. 6.1a is a conditional random field, the model in fig. 6.1b is often referred to as a hidden conditional random field (HCRF); see chapter 5. HCRFs are appropriate when there is a strong correlation between the observed coarse-grained variable and the unobserved fine-grained variables. We would expect to see positive, negative and neutral sentences in all types of documents, but we are far more likely to see positive sentences than negative sentences in positive documents.

Structured models with latent variables have previously been used for sentiment analysis, but only as a means to improve the prediction of the observed variables (Nakagawa et al., 2010; Yessenalina et al., 2010). Our approach differs in that we introduce hidden sentence level variables not only as a means to improve document sentiment predictions, but as a means for making meaningful predictions at the sentence level. As indicated in fig. 6.1b, there are two maximal cliques at each position i in the graphical model: one involving only the sentence x_i and its corresponding latent variable y_i^s and one involving the consecutive latent variables y_i^s, y_{i-1}^s and the document variable y^d.[4] The feature function thus factorizes as

$$\Phi(x, y) = \sum_{i=1}^{|x|} \phi(x_i, y_i^s) + \phi(y^d, y_i^s, y_{i-1}^s),$$

where $\phi(x_i, y_i^s)$ and $\phi(y^d, y_i^s, y_{i-1}^s)$ are the feature functions for the two cliques in fig. 6.1b.

According to this factorization, the assignment of the document variable y^d is independent of the input x, when conditioned on the sequence of latent sentence variables $\boldsymbol{y}^s$. This is in contrast to the original fine-to-coarse model of McDonald et al. (2007), in which the document variable depends directly on the sentence variables as well as on the input (as indicated by the larger maximal clique). This distinction is important for learning predictive latent variables as it creates a "bottleneck" between the input sentences and the document label, which forces the model to generate good predictions at the document level only through the predictions at the sentence level. Since the input x is highly informative of the document sentiment, the model might

[4] In chapter 5, we used z to refer to the latent variables. For ease of notation, in this chapter, we use y^s for the sentence variables both when they are observed and when they are latent. The correct interpretation should always be clear from context.

otherwise circumvent the latent sentence variables. Preliminary experiments confirmed this suspicion: when we allow the document label to be directly dependent on the input, we observe a substantial drop in sentence-level accuracy.

6.3.1 Learning and Inference

For training the models, we assume that we have access to a partially labeled training set $\mathcal{D}_p = \{(x^{(j)}, y^{d^{(j)}})\}_{j=1}^{m_p}$. While McDonald et al. (2007) used the MIRA algorithm (Crammer and Singer, 2003) — an online learning algorithm that optimizes a loss function similar to the structural SVM — for training their fully supervised model, we instead use the latent log loss (see section 5.2) optimized with stochastic gradient descent (see chapter 4). As in previous chapters, we train these models by minimizing the regularized empirical risk over the training set:

$$J(\theta; \mathcal{D}_p) = \frac{1}{m_p} \sum_{j=1}^{m_p} L(x^{(j)}, y^{d^{(j)}}, \theta) + \lambda \|\theta\|_2^2 ,$$

where $L(\cdot)$ is a surrogate loss function. In addition to the standard latent log loss, in which the latent variables are marginalized out, we also experimented with a hard-assignment variant of this loss, where the maximum a posteriori (MAP) assignment of the latent variables is used in place of marginalization. The latent log loss with soft assignment has the following form:

$$L_{\log}^{\text{soft}}(x, y^d, \theta) = -\left(\log Z_\theta(x, y^d) - \log Z_\theta(x)\right),$$

where

$$Z_\theta(x, y^d) = \sum_{y^s \in \mathcal{Y}_s(x)} \exp\left\{s_\theta(x, y^d, y^s)\right\}$$

is the partition function with x and y^d fixed and

$$Z_\theta(x) = \sum_{y \in \mathcal{Y}(x)} \exp\left\{s_\theta(x, y)\right\}$$

is the partition function with only x fixed. The latter is the same partition function as in the standard log loss. Similarly, the latent log loss with hard assignment has the following form:

$$L_{\log}^{\text{hard}}(x, y^d, \theta) = -\left(s_\theta(x, y^d, \hat{y}^s(y^d)) - \log \widehat{Z}_\theta(x)\right),$$

where the "MAP partition function" is given by

$$\widehat{Z}_\theta(x) = \sum_{y^{d'} \in \mathcal{Y}_d(x)} \exp\left\{s_\theta(x, y^{d'}, \hat{y}^s(y^{d'}))\right\}$$

and the MAP assignment of the sentence variables, treating y^d as fixed, is given by

$$\hat{y}^s(y^d) = \arg\max_{y^s \in \mathcal{Y}_s(x)} s_\theta(x, y^d, y^s)\,.$$

These models are subsequently referred to as HCRF (SOFT) and HCRF (HARD), respectively. We note that these models correspond to entropy regularization of the inferred distributions.

Recall that due to the non-convexity of these loss functions, any gradient-based optimization method is only guaranteed to find a local minimum of the objective function. Previous work on latent variable models for sentiment analysis by Nakagawa et al. (2010), and others, has reported on the need for complex initialization of the parameters to overcome the presence of local minima. We do not experience such problems and for all reported experiments we simply initialize θ to the zero vector.

At test time, when predicting the document and sentence-level sentiment, we can either use the global MAP assignment, or individually set each variable to the value with the highest marginal probability. It seems intuitively reasonable that the inference used at test time should match that used during training. Our experimental results indicate that this is indeed the case, although the differences between the decoding strategies are quite small.

Note that in the HCRF model the latent states assigned to the sentence variables, y_i^s, are not identifiable. We therefore need to find a mapping from the induced latent states to the labels that we are interested in, post-hoc. However, since the number of latent states is very small in our experiments, this mapping can be easily found by evaluating the possible mappings on a small set of annotated sentences. Alternatively, the HCRF may be seeded with values from the *DocAsSent* baseline, which directly fixes the assignment of latent variables to labels. Preliminary experiments suggest that this strategy always finds the optimal mapping.

6.3.2 Feature Templates

Below we list the feature templates used for the clique feature functions $\phi(x, y_i^s)$ and $\phi(y^d, y_i^s, y_{i-1}^s)$. The same features, with the exception for the structural features, are used for the SentAsDoc and DocAsSent baselines. While we only condition on the ith sentence in these features, since the model is discriminative, we could also condition on other parts of the input. We hypothesize that this could, for example, be potentially useful for capturing discourse, where certain words may signal sentiment shift.

Feature templates for the clique (x, y_i^s)*:*[5]

[5]In the present feature model, we ignore all sentences but x_i, so that instead of (x, y_i^s), we could have written (x_i, y_i^s). We keep to the more general notation; since the model is conditional, we could in principle look at any part of the input.

- $\textsc{Tokens}(x_i) \otimes y_i^s$
- $\textsc{NegatedTokens}(x_i) \otimes y_i^s$
- $\textsc{VoteFlip}(x_i) \otimes y_i^s$
- $\mathbb{1}\left[\textsc{NumPositive}(x_i) > \textsc{NumNegative}(x_i)\right] \otimes y_i^s$
- $\mathbb{1}\left[\textsc{NumPositive}(x_i) > 2 \cdot \textsc{NumNegative}(x_i)\right] \otimes y_i^s$
- $\mathbb{1}\left[\textsc{NumNegative}(x_i) > \textsc{NumPositive}(x_i)\right] \otimes y_i^s$
- $\mathbb{1}\left[\textsc{NumNegative}(x_i) > 2 \cdot \textsc{NumPositive}(x_i)\right] \otimes y_i^s$
- $\mathbb{1}\left[\textsc{NumPositive}(x_i) = \textsc{NumNegative}(x_i)\right] \otimes y_i^s$

Helper functions used for the above templates:

- $\textsc{Tokens}(x_i)$: The set of tokens in sentence x_i.
- $\textsc{NegatedTokens}(x_i)$: The set of tokens in sentence x_i that are in the scope of a negation.
- $\textsc{NumPositive}(x_i)$: The number of tokens in sentence x_i that are listed as positive in the lexicon.
- $\textsc{NumNegative}(x_i)$: The number of tokens in sentence x_i that are listed as negative in the lexicon.
- $\textsc{VoteFlip}(x_i)$: The output of the vote-flip algorithm (see algorithm 3) for sentence x_i.

All lexicon matches are against the MPQA polarity lexicon. The method described by Councill et al. (2010) is used to classify whether or not a token is in the scope of a negation.

In addition to these features, there is a simple set of structural feature templates for the clique (y^d, y_i^s, y_{i-1}^s), which only involve various combinations of the document- and sentence-sentiment variables.

Feature templates for the clique (y^d, y_i^s, y_{i-1}^s)*:*

- y^d
- y_i^s
- $y^d \otimes y_i^s$
- $y^d \otimes y_i^s \otimes y_{i-1}^s$

6.4 Experiments with Indirect Supervision

We now turn to a set of experiments in which we assess the viability of the proposed HCRF model compared to the *VoteFlip*, *SentAsDoc* and *DocAsSent* baselines described in section 6.2.

6.4.1 Experimental Setup

In order to make the underlying statistical models as similar as possible across systems, *SentAsDoc* and *DocAsSent* are also optimized with log loss using

stochastic gradient descent. These models thereby belong to the same model family as the HCRF model, except that they do not employ any latent variables to model document structure. With regards to the HCRF model, results are reported for both the soft and the hard variants of the latent log loss. Except where noted, results are reported using MAP inference for the hard model, and using marginal inference for the soft model. We also measure the benefit of observing the document label at test time. This is a common scenario in, for example, consumer-review summarization and aggregation (Hu and Liu, 2004a). Note that for this data set the baseline of predicting all sentences with the observed document label, denoted DocOracle, is a competitive baseline in terms of sentence-level sentiment accuracy. However, we shall see later that the HCRF models are much more informative.

The *SentAsDoc*, *DocAsSent* and HCRF models all depend on three hyper-parameters during training: the stochastic gradient descent learning rate η, the regularization trade-off parameter λ, and the number of training epochs; see section 4.1.1 and section 4.1.4. We allow a maximum of 75 epochs and pick values for the hyper-parameters that maximize macro-averaged F_1 on the document level for all models, as measured on a separate development data set. No manual sentence-level supervision is used during any point of training for any of the models. Sentence-level annotations are only used in order to identify the latent states when evaluating the trained models. As discussed above, the latent states could also be identified by using the *DocAsSent* model. The three models use identical feature sets when possible (see section 6.3.2), the single exception being that *SentAsDoc* and *DocAsSent* do not use any structural features, such as adjacent sentence label features, since they are not structured predictors. For all models, a 19-bit hash kernel is used to map the feature template instantiations to feature space elements; see section 4.2.3. Except for the lexicon-based model, the training of all models is stochastic in nature. To account for this, we train each model ten times, each time with a different random seed. In each training run a different split of the training data is used for tuning the hyper-parameters. We then gather the results by applying each model to the test data described in section 6.1 and bootstrapping the median and 95% confidence intervals of the statistic of interest (Efron and Tibshirani, 1993). Since the iid assumption cannot reasonably be assumed to hold for sentence-level sentiment predictions, due to intra-document dependencies, we use a hierarchical bootstrap (Davison and Hinkley, 1997).

6.4.2 Results and Analysis

Table 6.5 shows the results for each model in terms of sentence-level and document-level accuracy, as well as macro-averaged F_1-scores for each sentence sentiment category. Results are given as median accuracy and F_1-score, with accuracy scores supplemented by their 95% bootstrapped confidence in-

Table 6.5. *Sentence- and document-level results for the large data set. The bootstrapped median result and 95% confidence interval is shown for each model. Above mid-line: without observed document label. Below mid-line: with observed document label. Bold-faced: Statistically significant compared to the best comparable baseline, according to a hierarchical bootstrap test ($p < 0.05$).*

	Sentence Accuracy	POS Sent. F_1	NEG Sent. F_1	NEU Sent. F_1	Document Accuracy
VoteFlip	41.5 (-1.8, 1.8)	45.7	48.9	28.0	–
SentAsDoc	47.6 (-0.8, 0.9)	52.9	48.4	42.8	–
DocAsSent	47.5 (-0.8, 0.7)	52.1	54.3	36.0	66.6 (-2.4, 2.2)
HCRF (soft)	**53.9 (-2.4, 1.6)**	**57.3**	**58.5**	**47.8**	65.6 (-2.9, 2.6)
HCRF (hard)	**54.4 (-1.0, 1.0)**	**57.8**	**58.8**	**48.5**	64.6 (-2.0, 2.1)
DocOracle	54.8 (-3.0, 3.1)	61.1	58.5	47.0	–
HCRF (soft)	**57.7 (-0.9, 0.8)**	61.5	**62.0**	**51.9**	–
HCRF (hard)	**58.4 (-0.8, 0.7)**	62.0	**62.3**	**53.2**	–

terval. From these results it is clear that the HCRF models significantly outperform all the baselines with quite a wide margin. When document labels are provided at test time, results are even better compared to the machine learning baselines, but compared to the *DocOracle* baseline, the error reductions are more modest. These differences are all statistically significant at $p < 0.05$ according to a (hierarchical) bootstrap test.

Specifically, in terms of relative error reduction, the HCRF with hard estimation reduces the error compared to the pure lexicon approach by 22% and by 13% compared to the best machine learning baseline. When document labels are provided at test time, the corresponding error reductions are 29% and 21%. In the latter case the reduction compared to the strong *DocOracle* baseline is only 8%. However, as we shall see subsequently, the probabilistic predictions of the HCRF are much more informative than this simple baseline. On average, hard estimation for the HCRF slightly outperforms soft estimation; however, this difference is not statistically significant.

In terms of document accuracy, the *DocAsSent* model seems to slightly outperform the latent variable models (again the difference is not statistically significant). This is contrary to the results reported in Yessenalina et al. (2010), in which latent variables on the sentence level was shown to improve document predictions. Note, however, that our model is restricted when it comes to document level classification, due to the lack of connection between the document variable and the input in the graphical model. If we let the document sentiment be directly dependent on the input, which is similar to a probabilistic formulation of the max-margin approach proposed by Yessenalina et al. (2010), we would expect the document accuracy to improve. However, experiments with such connected HCRF models actually showed

Table 6.6. *Sentence-level sentiment classification results per document category (columns). Each cell contains* POS / NEG / NEU *sentence-level* F_1*-scores.*

	POS documents	NEG documents	NEU documents
VoteFlip	59 / 19 / 27	16 / 61 / 23	40 / 51 / 32
SentAsDoc	67 / 18 / 45	15 / 60 / 36	43 / 42 / 45
DocAsSent	67 / 20 / 35	14 / 68 / 29	45 / 49 / 41
HCRF (soft)	69 / 14 / 45	7 / 70 / 37	33 / 49 / 55
HCRF (hard)	69 / 14 / 47	6 / 71 / 36	34 / 48 / 56
DocOracle	69 / 0 / 0	0 / 77 / 0	0 / 0 / 67
HCRF (soft)	70 / 1 / 39	2 / 76 / 29	20 / 36 / 66
HCRF (hard)	72 / 0 / 44	0 / 76 / 23	3 / 38 / 66

a slight decrease in document level accuracy compared to the disconnected models, while sentence level accuracy dropped even below the *SentAsDoc* and *DocAsSent* models. By initializing the HCRF models with the parameters of the *DocAsSent* model, better results were obtained, but still not on par with the disconnected models.

Looking in more detail at table 6.5, we observe that all models perform best in terms of F_1-score on positive and negative sentences, while all models perform much worse on neutral sentences. This is not surprising, since neutral documents are particularly bad proxies for sentence-level sentiment, as can be seen from the distributions of sentence-level sentiment per document category in table 6.3. The lexicon based approach has difficulties with neutral sentences, since the lexicon contains only positive and negative words and there is no way of determining if a mention of a word in the lexicon should be considered as carrying sentiment in a given context.

A shortcoming of the HCRF model, compared to the baselines, is suggested by table 6.6: the former tends to over-predict positive sentences in positive documents (and analogously for negative sentences in negative documents) and to under-predict positive sentences in neutral documents. In other words, it only works well for the two dominant sentence-level categories for each document category. This is a problem shared by the baselines, but it is more prominent in the HCRF model. A plausible explanation resides in the fact that we are optimizing the ability to predict document-level sentiment; in order to learn whether a review is positive, negative or neutral, it will often suffice to find the dominant sentence-level sentiment and to identify the non-relevant sentences of the review. Therefore, the model might need more constraints in order to learn to predict the minority sentence-level sentiment categories. In section 6.5, we show that a semi-supervised approach provides a partial solution to these issues.

Table 6.7. *Sentence-level sentiment accuracy with training sets of varying sizes. Number of documents used for training:* Small*:* 1500*;* Medium*:* 15 000*; and* Large*:* 143 580*. The bootstrapped median result and 95-percent confidence interval is shown for each model. Bold: Statistically significant compared to all comparable baselines, according to a bootstrap test ($p < 0.05$).*

	Small	Medium	Large
VoteFlip	41.5 (-1.8, 1.8)	41.5 (-1.8, 1.8)	41.5 (-1.8, 1.8)
SentAsDoc	42.4 (-2.0, 1.3)	46.3 (-1.2, 1.0)	47.6 (-0.8, 0.9)
DocAsSent	43.8 (-0.9, 0.8)	46.8 (-0.6, 0.7)	47.5 (-0.8, 0.7)
HCRF (soft)	44.9 (-1.7, 1.5)	**50.0 (-1.2, 1.2)**	**53.9 (-2.4, 1.6)**
HCRF (hard)	43.0 (-1.2, 1.3)	**49.1 (-1.4, 1.5)**	**54.4 (-1.0, 1.0)**
DocOracle	54.8 (-3.0, 3.1)	54.8 (-3.0, 3.1)	54.8 (-3.0, 3.1)
HCRF (soft)	54.5 (-1.0, 0.9)	54.9 (-1.0, 0.8)	**57.7 (-0.9, 0.8)**
HCRF (hard)	48.6 (-1.6, 1.4)	54.3 (-1.9, 1.8)	**58.4 (-0.8, 0.7)**

The impact of more data

In order to study the impact of varying the size of the training data, we create additional training sets, denoted *Small* and *Medium*, by sampling 1500 and 15 000 documents, respectively, from the full training set, denoted *Large*. We then perform the same experiment and evaluation as with the full training set with these smaller sets. As in the previous experiments, different training set samples are used for each run of the experiment. From table 6.7, we observe that adding more training data consistently improves all models. For the small data set, there is no significant difference between the learning-based models, but starting with the medium data set, the HCRF models outperform the baselines. Furthermore, while the improvement from adding more data is relatively small for the baselines, the improvement is substantial for the HCRF models. We therefore expect that the gap between the latent variable models and the baselines will continue to increase with increasing training set size. Training sets with millions of product reviews are not inconceivable. There are no significant technical barriers to training our models on such large data sets, although we may be forced to employ parallelized learning algorithms; see section 4.2.1.

Trading off precision against recall

Although MAP inference slightly outperforms marginal inference for the hard HCRF in terms of classification performance, using marginal inference for prediction has the advantage that we can tune per-label precision versus recall based on the sentence-level marginal distributions. This flexibility is yet another reason for preferring statistical approaches to rule-based approaches, such as *VoteFlip* and the *DocOracle* baseline. Figure 6.2 shows sentence-level precision–recall curves for the HCRF models (with and without observed document labels), *SentAsDoc* and *DocAsSent*, together with

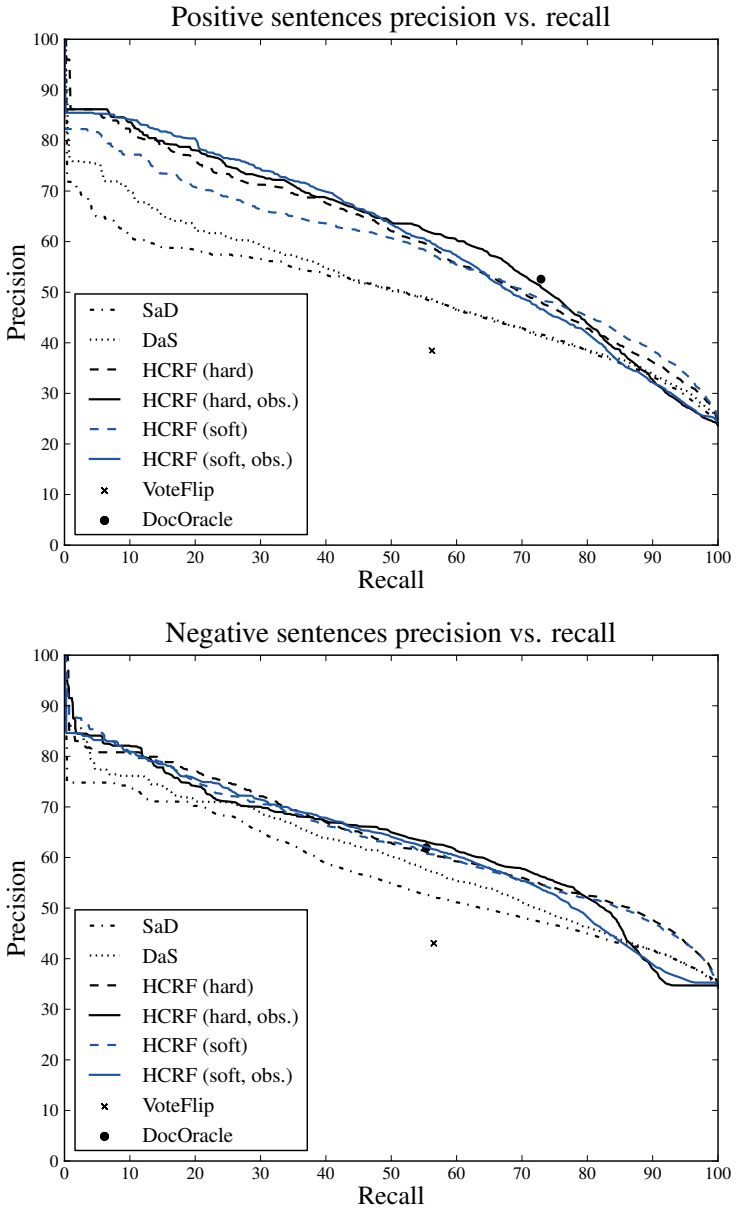

Figure 6.2. Interpolated precision–recall curves with respect to positive and negative sentence-level sentiment. *SaD*: SentAsDoc. *DaS*: DocAsSent. Each curve corresponds to the bootstrapped median of average precision over ten runs.

Table 6.8. *Sentence- and document-level results for the large data set with* neutral documents excluded. *The bootstrapped median result and 95-percent confidence interval is shown for each model. Above line: without observed document label. Below line: with observed document label. Boldfaced: Statistically significant compared to best comparable baseline, according to a bootstrap test ($p < 0.05$).*

Method	Sentence Accuracy	POS Sent. F_1	NEG Sent. F_1	NEU Sent. F_1	Document Accuracy
VoteFlip	41.5 (-1.9, 2.0)	48.2	47.7	25.0	–
SentAsDoc	49.0 (-1.2, 1.2)	57.7	59.7	11.1	–
DocAsSent	48.3 (-0.9, 0.9)	57.3	60.7	0.0	87.5 (-1.5, 1.6)
HCRF (soft)	**57.6 (-1.3, 1.2)**	**63.6**	**66.9**	**39.4**	88.4 (-1.9, 1.6)
HCRF (hard)	**53.7 (-1.5, 1.7)**	**62.8**	**68.8**	0.0	87.8 (-1.5, 1.5)
DocOracle	57.3 (-4.0, 3.6)	67.1	72.5	–	–
HCRF (soft)	**60.6 (-1.0, 1.0)**	68.2	71.5	**38.2**	–
HCRF (hard)	57.6 (-1.4, 1.6)	66.2	71.7	16.0	–

the fix points of *VoteFlip* and *DocOracle*. Each is formed from the bootstrapped median of precision for each recall level, computed over ten runs with different random seeds. Based on these plots, it is evident that the HCRF models dominate the other models in terms of sentence-level predictions at nearly all levels of recall, in particular for positive sentences.

Ignoring neutral documents

Although the results seem low across the board — below 60% sentence-level accuracy and below 70% document-level accuracy — they are comparable with those of McDonald et al. (2007), who report 62.6% sentence-level accuracy for a model trained with both document- and sentence-level supervision, and evaluated on a data set that did not contain neutral documents. In fact, the primary reason for the low scores presented in this work is the inclusion of neutral documents and sentences in our data. This makes the task much more difficult than 2-class positive-negative polarity classification, but also more representative of real-world use-cases.

To support this claim, we perform the same experiments as above while excluding neutral documents from the training and test data. Table 6.8 contains detailed results for the two-class experiments, while fig. 6.3 shows the corresponding precision–recall curves. In this scenario the best HCRF model achieves a document accuracy of 88.4%, which is roughly on par with reported document accuracies for the two-class task in state-of-the-art systems (Blitzer et al., 2007; Nakagawa et al., 2010; Yessenalina et al., 2010). As mentioned in section 6.1, inter-annotator agreement is only 86% for the three-class problem, which can be viewed as an upper bound on sentence-level accuracy. Interestingly, while excluding neutral documents improve accuracies and F_1-scores of positive and negative sentences, which is not unexpected

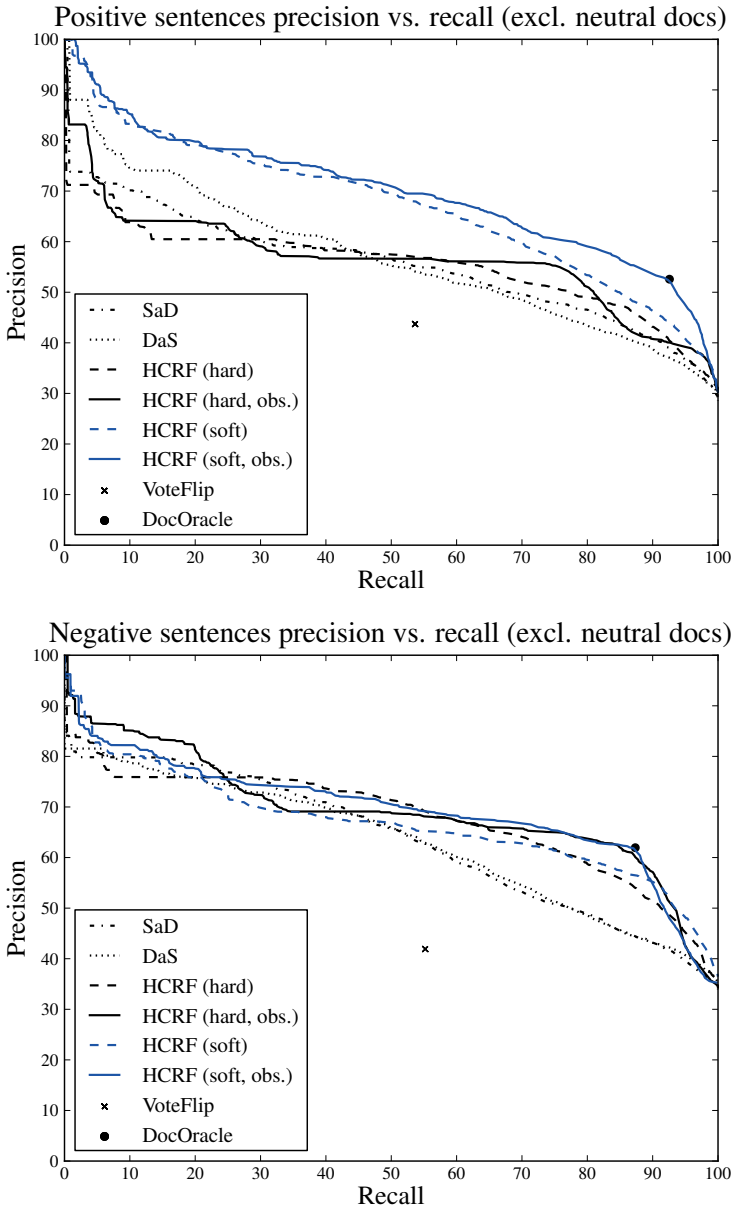

Figure 6.3. Interpolated precision–recall curves with respect to positive and negative sentence-level sentiment with *neutral documents excluded*. *SaD*: SentAsDoc. *DaS*: DocAsSent. Each curve corresponds to the bootstrapped median of average precision over ten runs.

since the task is made simpler, F_1-scores for neutral sentences are much lower. In the *DocAsSent* and hard HCRF cases, the models completely fail to predict any neutral sentence-level sentiment. This is not surprising, since we are learning solely from positive and negative document-level sentiment in these experiments. Nevertheless, the soft HCRF model is able to learn to predict neutral sentences to some degree even in this case.

6.5 Two Semi-Supervised Models

While the HRCF model surpasses a set of natural baselines with quite a wide margin, it has its shortcomings, as highlighted in our empirical study. Most notably, due to the loose constraints provided by the document-level supervision, it tends to only predict the two dominant sentence-level sentiment categories well for each document-level sentiment category. That is, it deems almost all sentences in positive documents as positive or neutral, and similarly for negative documents. As a way of overcoming these shortcomings, we propose two semi-supervised variants of the HCRF model, both of which are based on a combination of the partially supervised HCRF model and the fully supervised fine-to-coarse model of McDonald et al. (2007). In addition to the large partially labeled training set $\mathcal{D}_p = \{(x^{(j)}, y^{d(j)})\}_{j=1}^{m_p}$, we assume that we have access to a training set of fully labeled instances $\mathcal{D}_f = \{(x^{(j)}, y^{(j)})\}_{j=1}^{m_f}$. Below, we first describe the two model variants; we then verify experimentally that they both yield significantly improved sentence-level sentiment predictions, compared to all baselines.

6.5.1 A Cascaded Model

A straightforward way of fusing the partially supervised and the fully supervised models is by means of a cascaded model, where the predictions of the partially supervised model are used to derive additional features for the fully supervised model. As before, the former is trained by minimizing the latent log loss, while the latter is trained by minimizing the standard log loss. Subsequently, let θ_p denote the parameters of the partially supervised model and let θ_f denote the parameters of the fully supervised model. Denote the corresponding objective functions by $J_p(\theta_p; \mathcal{D}_p)$ and $J_f(\theta_f; \mathcal{D}_f)$, respectively.

While more complex schemes are possible, the meta-features generated for each sentence are based solely on operations on the estimated conditional distributions $p_{\theta_p}(y^d, y_i^s \mid x)$. For each sentence x_i, we encode the following distributions as discrete features by uniform bucketing into $\{0, 0.1, 0.2, \ldots, 1.0\}$: the joint distribution $p_{\theta_p}(y^d, y_i^s \mid x)$, the marginal document-level distribution $p_{\theta_p}(y^d \mid x)$, and the marginal sentence-level distribution $p_{\theta_p}(y_i^s \mid x)$. The MAP assignments of these distributions are also encoded; that is, for each dis-

Table 6.9. *Sentence-level results for varying numbers of fully labeled ($\mathcal{D}_f$) and partially labeled ($\mathcal{D}_p$) reviews. The bootstrapped median result and 95-percent confidence interval is shown for each model. Bold: significantly better than the* FineToCoarse *model according to a bootstrap test ($p < 0.05$).*

	$\left\|\mathcal{D}_p\right\| = 15\,000$		
	$\left\|\mathcal{D}_f\right\| = 60$	$\left\|\mathcal{D}_f\right\| = 120$	$\left\|\mathcal{D}_f\right\| = 240$
FineToCoarse	49.3 (-1.3, 1.4)	53.4 (-1.8, 1.7)	54.6 (-3.6, 3.8)
HCRF (soft)	49.6 (-1.5, 1.8)	49.6 (-1.5, 1.8)	49.6 (-1.5, 1.8)
Cascaded	39.7 (-6.8, 5.7)	45.4 (-3.1, 2.9)	42.6 (-6.5, 6.5)
Interpolated	**54.3 (-1.4, 1.4)**	**55.0 (-1.7, 1.6)**	**57.5 (-4.1, 5.2)**

	$\left\|\mathcal{D}_p\right\| = 143\,580$		
	$\left\|\mathcal{D}_f\right\| = 60$	$\left\|\mathcal{D}_f\right\| = 120$	$\left\|\mathcal{D}_f\right\| = 240$
FineToCoarse	49.3 (-1.3, 1.4)	53.4 (-1.8, 1.7)	54.6 (-3.6, 3.8)
HCRF (soft)	**53.5 (-1.2, 1.4)**	53.5 (-1.2, 1.4)	53.5 (-1.2, 1.4)
Cascaded	**55.6 (-2.9, 2.7)**	55.0 (-3.2, 3.4)	**56.8 (-3.8, 3.6)**
Interpolated	**56.0 (-2.4, 2.1)**	54.5 (-2.9, 2.8)	**59.1 (-2.8, 3.4)**

tribution $p(e \mid x)$ we also encode $\arg\max_{e \in \mathcal{E}} p(e \mid x)$ as a feature. In the case where the document labels are observed at test time, we additionally encode the sentence-level distribution conditioned on the observed document label $p_{\theta_p}(y_i^s \mid x, y^d)$, as well as the MAP assignment of this conditional distribution. Finally, all pairwise combinations of the above features are also added to the feature set.

The upshot of this cascaded approach is that it is very simple to implement and efficient to train. The downside is that only the partially supervised model influences the fully supervised model; there is no reciprocal influence between the models. Given the non-concavity of the latent log loss, such influence could be beneficial, as the fully labeled data provides the strongest possible constraints on the sentence-level variables. Furthermore, although the features derived from the partially supervised model are much less sparse, compared to the other features in the fully supervised model, the former are also less flexible and the sentence-level signal is still restricted as the fully labeled training set remains small, which makes it prone to overfit.

6.5.2 An Interpolated Model

A more flexible way of fusing the two models is to interpolate their respective objective functions. This facilitates the direct combination of document-level and joint supervision in a single model. Such a combination can be easily achieved by constraining the parameters, such that $\theta = \theta_f = \theta_p$, and by

interpolating the objective functions $J_f(\theta; \mathcal{D}_f)$ and $J_p(\theta; \mathcal{D}_p)$, appropriately weighted by a hyper-parameter α:

$$J(\theta; \mathcal{D}_f, \mathcal{D}_p) \; = \alpha J_f(\theta; \mathcal{D}_f) + (1 - \alpha) J_p(\theta; \mathcal{D}_p)\,.$$

A straightforward way of optimizing this objective function is to use stochastic gradient descent with learning rate η. At each step, we pick a fully labeled instance $(x, y) \sim \mathcal{D}_f$ with probability α and, consequently, we pick a partially labeled instance $(x, y^d) \sim \mathcal{D}_p$ with probability $(1 - \alpha)$. We then update the parameters θ according to the gradients $\frac{\partial}{\partial\theta} J_f(\theta; \mathcal{D}_f)$ and $\frac{\partial}{\partial\theta} J_p(\theta; \mathcal{D}_p)$, respectively. In principle we could use different learning rates η_f and η_p as well as different regularization hyper-parameters λ_f and λ_p, but in what follows we set them equal. Note that since both component models share the same parameters, they need to use the same features and identical graphical models, that is, the one outlined in fig. 6.1b.

6.6 Experiments with Semi-Supervision

We next compare the two proposed semi-supervised models (Cascaded and Interpolated) to the supervised fine-to-coarse model (FineToCoarse), as well as to the soft indirectly supervised HCRF model (HCRF (soft)).

6.6.1 Experimental Setup

For these experiments, we fix the learning rate of stochastic gradient descent to $\eta = 0.001$, while we tune the regularization hyper-parameter λ using cross-validation on the training set. Each model is trained for a maximum of 30 epochs. When sampling according to α during optimization of $J(\theta; \mathcal{D}_f, \mathcal{D}_p)$, we cycle through $\mathcal{D}_f$ and $\mathcal{D}_p$ deterministically, while shuffling the sets between epochs. For simplicity, we fix the interpolation factor to $\alpha = 0.1$; tuning this could potentially improve the results of the interpolated model.

In order to assess the impact of fully labeled versus partially labeled data, we took stratified samples without replacement of 60, 120, and 240 reviews, respectively, from the fully labeled data, and of sizes 15 000 and 143 580 reviews from the partially labeled data. For evaluation, we perform a 5-fold stratified cross-validation over the fully labeled data set, while using stratified samples of the partially labeled data. As in the previous experiments, statistical significance is assessed by a hierarchical bootstrap test.

6.6.2 Results and Analysis

Table 6.9 lists sentence-level median accuracies along with the 95% bootstrapped confidence interval for all tested models. From these results, we

Table 6.10. *POS / NEG / NEU sentence-level F_1-scores per document sentiment category, with $|\mathcal{D}_p|$ = 143 580 and $|\mathcal{D}_f|$ = 240.*

	POS documents	NEG documents	NEU documents
FINETOCOARSE	35 / 11 / 59	33 / 76 / 42	29 / 63 / 55
HCRF (SOFT)	70 / 14 / 43	11 / 71 / 34	43 / 47 / 53
CASCADED	43 / 17 / 61	0 / 75 / 49	10 / 64 / 50
INTERPOLATED	73 / 16 / 51	42 / 72 / 48	54 / 52 / 57

observe that the interpolated model dominates all other models in terms of accuracy. Comparing the two semi-supervised models, we see that while the cascaded model requires both large amounts of fully labeled and partially labeled data, the interpolated model is able to take advantage of both types of data on its own and jointly. Comparing the fully supervised and the partially supervised models, the superior impact of fully labeled over partially labeled data is evident. Turning to the precision–recall curves in figs. 6.4 and 6.5, when all data is used, the cascaded model outperforms the interpolated model for some recall values, and vice versa. Both models dominate the supervised approach for the full range of recall values, with the exception of high recall values, in the case of observed document labels as seen from fig. 6.5; in this case the purely supervised model achieves higher precision compared to the interpolated model.

As discussed earlier, and confirmed again by the results in table 6.10, the partially supervised model only performs well on the predominant sentence-level categories for each document category. The supervised model handles negative and neutral sentences well, but perform poorly on positive sentences even in positive documents. The interpolated model, while still better at capturing the predominant category, achieves higher F_1-scores overall.

6.7 Discussion

Structured latent variable models for sentiment analysis have recently been investigated in related work. Nakagawa et al. (2010) presented a sentence-level model where the observed information is the polarity of a sentence and the latent variables correspond to nodes in the syntactic dependency tree for the sentence. They showed that such a model can improve sentence-level polarity classification, when the sentence-level polarity is observed during training. Yessenalina et al. (2010) presented a document-level model where the latent variables are binary predictions over sentences. These variables indicate whether the sentence should be considered when classifying the

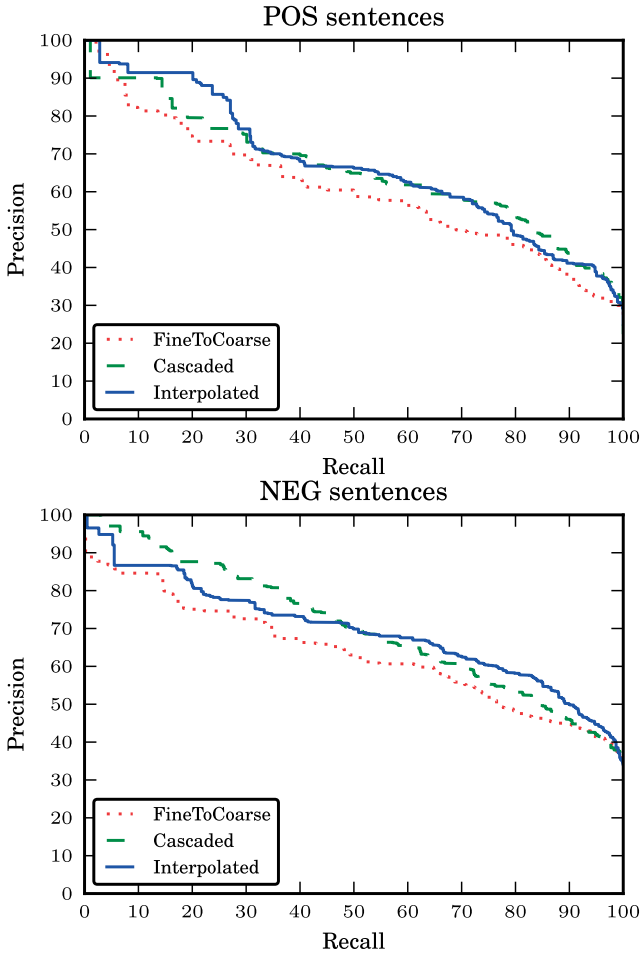

Figure 6.4. Interpolated precision–recall curves with respect to positive and negative sentence-level sentiment.

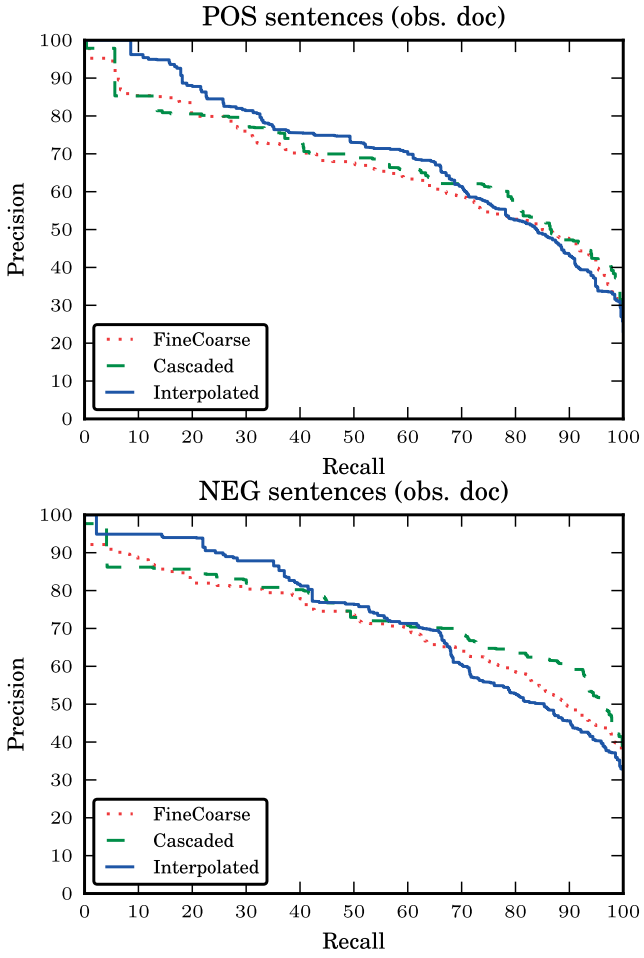

Figure 6.5. Interpolated precision–recall curves with respect to positive and negative sentence-level sentiment with the *document label observed at test time.*

document or if it should be disregarded.[6] In both of these approaches, the primary goal is to improve the performance of the model on the supervised annotated signal. The latent variables themselves are never used for prediction, although the authors suggest that the variables should correlate with the sub-sentence or sub-document sentiment of the text under consideration.

In this chapter, we instead focus on using latent variables directly for prediction and we invert the evaluation in an attempt to assess the accuracy of the latent structure induced from the observed coarse supervision. In fact, one could argue that learning fine-grained sentiment from document-level labels is the more relevant question for multiple reasons. First, document-level annotations are the most common naturally observed sentiment signal, for example, in the form of consumer reviews with ratings. Second, fine-grained annotations often require large annotation efforts (Wiebe et al., 2005), which have to be undertaken on a domain-by-domain basis. Third, document-level sentiment analysis is too coarse-grained for most sentiment applications, especially those that rely on aggregation across fine-grained topics (Hu and Liu, 2004a).

Recent work by Chang et al. (2010) had the similar goal of inducing latent structure from binary indirect supervision indicating whether each instance (in our case, each review) is valid or not, though they did not specifically investigate sentiment analysis. As discussed in chapter 5, their model is a special case of the structured latent variable model presented in this chapter. The generalization to non-binary labels is important, since there is no natural notion of invalid instances in the sentiment analysis task and since the model needs to distinguish between multiple sentiment categories.

After the original publication of this work, Qu et al. (2012) proposed a similar model based on learning with multiple experts. They include a graph-based regularizer, which makes use of sentence similarity to propagate information from sentences that are likely to be correctly classified by a base predictor, for example a lexicon, to similar sentences for which the base predictor is less confident. Empirically, this more complex model was shown to outperform the authors' reimplementation of the indirectly supervised and semi-supervised models presented in this chapter.

Since we are interpolating (or cascading) discriminative models, we need at least partial observations of each instance. This is in contrast to generative topic models, which have been amply used for unsupervised (Mei et al., 2007; Titov and McDonald, 2008; Lin and He, 2009) and weakly supervised sentiment analysis (He, 2011; Lin et al., 2012). Methods for blending discriminative and generative models (Lasserre et al., 2006; Suzuki et al., 2007; Agarwal and Daumé, 2009; Sauper et al., 2010), would enable the incorporation of additional fully unlabeled data. It is certainly possible to extend the proposed

[6] While the original report on this work (Täckström and McDonald, 2011a) was published after that of Yessenalina et al. (2010), our work was performed independently.

model along these lines. However, in practice, partially labeled product review data is so abundant on the web that incorporating unlabeled data seems superfluous in our setting. Furthermore, the proposed discriminative models with shared parameters allow rich overlapping features, while inference and estimation remain simple and efficient.

Finally, we note that the approach that we propose in this chapter is orthogonal to semi-supervised and unsupervised induction of prior polarity lexica (Turney, 2002; Kim and Hovy, 2004; Esuli and Sebastiani, 2006a; Rao and Ravichandran, 2009; Velikovich et al., 2010). The output of such models could readily be incorporated as features in the proposed model.

Part III:
Learning with Cross-Lingual Supervision

7. Learning with Cross-Lingual Supervision

The previous two chapters described learning with different types of incomplete supervision. While general in nature, the empirical study of this setting in the previous chapter was restricted to English data. This chapter, instead considers the prediction of linguistic structure in the multilingual setting. In particular, an overview is provided of different ways of sharing and transferring linguistic knowledge across languages. This chapter provides the prerequisites for chapters 8 to 10, where we develop novel approaches to learning with cross-lingual supervision.

7.1 Multilingual Structure Prediction

As discussed in chapter 1 and chapter 5, access to core natural language processing tools is still lacking for most languages, due to the reliance on fully supervised learning methods, which require large quantities of manually annotated training data. Most natural language processing research has consequently been focused on Indo-European and East Asian languages and in particular on English, since it is the language for which most annotated resources are available. Notable exceptions include the CoNLL shared tasks (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003; Buchholz and Marsi, 2006; Nivre et al., 2007) and subsequent studies on this data, as well as a number of focused studies on one or two specific languages, as discussed by Bender (2011); see also the proceedings of the SPMRL workshops (Seddah et al., 2010, 2011). While annotated resources for syntactic parsing and several other tasks are available in a number of languages, we cannot expect to have access to fully annotated resources for all tasks in all languages any time soon. Hence, we need to explore alternatives to methods that rely on full supervision in each target language.

In chapter 5, we discussed methods for overcoming this hurdle by leveraging incomplete supervision, or no ostensive supervision at all. However, although partial or ambiguous supervision may be acquired for some types of linguistic structure, for example, sentiment (see chapter 6) and parts of speech (see chapter 10), such supervision is in many cases not naturally available. Unsupervised methods are therefore appealing, since they do not even require incomplete supervision and since they are often inherently language independent (although the results of a particular unsupervised learning method often vary substantially across languages). This is borne out by, for example,

the many recent studies on unsupervised part-of-speech tagging and syntactic parsing that include evaluations covering a number of languages (Cohen and Smith, 2009; Gillenwater et al., 2010; Naseem et al., 2010; Spitkovsky et al., 2011, 2012; Tu and Honavar, 2012). However, the performance of unsupervised methods is still substantially below that of supervised systems and recent work has established that the performance is also well below simple methods of *cross-lingual learning* (McDonald et al., 2011; Das and Petrov, 2011), which are the focus of this chapter.

The rationale for cross-lingual learning is that, rather than starting from scratch when creating a linguistic processing system for a resource-poor target language, we should take advantage of any corresponding annotation that is available in one or more resource-rich languages. Typically, this is achieved either by projecting annotations, or by transferring models, from source language(s) to target language(s). Recently, such methods have appeared as a promising route to overcoming the lack of annotated data in, for example, part-of-speech tagging (Yarowsky et al., 2001; Das and Petrov, 2011) and syntactic dependency parsing (Hwa et al., 2005; Ganchev et al., 2009; McDonald et al., 2011; Naseem et al., 2012) and named-entity recognition (Kim et al., 2012). While these methods do not reach up to the accuracy of fully supervised approaches, they have been shown to drastically outperform both unsupervised methods (Klein and Manning, 2004) and methods that learn from weak constraints (Naseem et al., 2010; Berg-Kirkpatrick and Klein, 2010).

Next, we briefly characterize different multilingual learning scenarios and we discuss a variety of cross-lingual resources and their use in cross-lingual learning. In section 7.2, we discuss and compare the two primary means for cross-lingual learning in more detail. This is followed by a brief discussion of learning with multiple source languages and some additional cross-lingual learning methods. The chapter ends with a discussion of the difficulties involved in evaluating linguistic processing systems in the cross-lingual setting.

7.1.1 Multilingual Learning Scenarios

There are essentially three different scenarios to linguistic structure prediction when taking multilinguality into account: either supervision is available in all languages of interest, supervision is available in none of the languages, or supervision is available only in a subset of the languages. We will subsequently focus on the latter scenario, but let us first briefly discuss these scenarios, focusing in particular on methods for leveraging cross-lingual relationships between the languages to guide learning.

A theme common to these approaches is that different constructions in different languages provide the model with constraints that can guide learning.

An often cited example, in the context of multilingual syntactic parsing,[1] is the fact that while PP-attachment — the problem of disambiguating which part of an utterance a prepositional phrase modifies, for example, where to attach *with the binoculars* in *She saw the man with the binoculars* — is notoriously difficult in English and other Indo-European languages, it is trivial and non-ambiguous in, for example, Chinese (Fossum and Knight, 2008; Chen et al., 2010) and Urdu (Snyder et al., 2009). Therefore, when parsing an English sentence, if a translation of the sentence is available to a language which is non-ambiguous in this respect, one could leverage the syntactic structure of the translated sentence to inform the parsing of the English sentence. Since different constructions tend to be ambiguous in different languages, jointly inferring the syntactic structure of two (or more) languages can be advantageous.

To this end, generative bilingual models have been proposed, such as the inversion transduction grammars (ITGs) introduced by Wu (1997), and the stochastic bilingual multitext grammars (2-MTGs) of Melamed (2003), which were used for joint bilingual parsing by Smith and Smith (2004). Another variant is the hierarchical phrase-based model of Chiang (2007). Burkett et al. (2010) proposed a weakly synchronized discriminative model, where the syntactic structure of two languages is modeled jointly with an alignment between the structures. In contrast to ITGs and 2-MTGs, which tightly couple the two syntactic structures, the structures are rather encouraged to agree with each other via the alignment. Inference in this model is performed with a variational mean-field approximation to the full joint model. Unfortunately, the computational complexity of parsing with these joint bilingual models is prohibitively high, for example, $\mathcal{O}(n^8)$ for 2-MTGs and $\mathcal{O}(n^6)$ for ITGs, ignoring large grammar constants.[2] Focus has therefore been on approximate inference (Burkett et al., 2010) and on reranking methods (Burkett and Klein, 2008). An even more common approach is to take the structure of one language as fixed and given, and use this to inform the prediction of structure in the other language. This is the approach taken in subsequent chapters, as our aim is to create systems that can be applied to monolingual text, while the models discussed above are only applicable to parsing of *bitext*, that is, to pairs of sentences that are translations of each other.

All languages supervised

In the most trivial — and practically implausible — scenario, full supervision is available for all languages of interest. This scenario has been studied, for example, in the CoNLL shared tasks, in which supervised named-entity recognizers were evaluated across four Indo-European languages (Tjong Kim Sang,

[1] In this section we focus on syntactic parsing, but similar methods may be applicable to other tasks as well.

[2] Inference in linear transduction grammars (LTGs), a restricted linearized variant of ITGs, can be performed in $\mathcal{O}(n^4)$ time (Saers, 2011).

2002; Tjong Kim Sang and De Meulder, 2003) and supervised syntactic dependency parsers across 19, primarily Indo-European, languages (Buchholz and Marsi, 2006; Nivre et al., 2007). The results of these studies show a rather large spread in performance across languages. Learning to predict linguistic structure in multiple languages more generally remains a challenge even when full supervision is available for each language, since different languages often benefit from different features, models and algorithms. While tools such as the automatic model and feature optimizer for syntactic dependency parsing of Ballesteros and Nivre (2012) can be used to at least partially automate the process of tuning these aspects to each language, developing truly multilingual systems remains an open challenge, as discussed at length by Bender (2011). Part of the challenge is that certain phenomena are intrinsically more difficult to analyze in some languages, such as PP-attachment in Indo-European languages, word segmentation in Chinese and sentence-boundary detection in Thai, as discussed above and in section 2.1.

Despite the fact that fully supervised systems typically constitute highly competitive baselines, given enough labeled training data, this scenario has been considered in a number of recent studies of bilingual syntactic parsing (Smith and Smith, 2004; Burkett and Klein, 2008; Fossum and Knight, 2008; Huang et al., 2009; Fraser et al., 2009; Burkett et al., 2010; Chen et al., 2010; Haulrich, 2012). A primary motivation behind this work is to improve syntax-based machine translation systems, where one needs to infer the syntactic structure, as well as alignments between these structures, for a pair of parallel sentences. Another common motivation is to enable the use of resource-rich languages with large amounts of fully labeled data to guide the learning of less resource-rich languages.

No language supervised

In the converse of the previous scenario, supervision is available for none of the languages for which the model should be applied. This setting has been considered in a number of recent studies, where it has been shown that leveraging cross-lingual relationships in the unsupervised learning process can provide substantial improvements, compared to treating each language separately.

Many of these approaches are based on the idea of decomposing the model into one component which is shared between all languages and a language specific component for each language. During learning, the model is then encouraged to use the shared part to explain as many phenomena as possible in the different languages, so that the language specific parts of the model are only used when required to explain cross-lingual divergences. The assumption is thus that the surface realizations of different languages to a large extent are bound by regularities that hold universally across the world's languages. This assumption is often entrenched in the model structure, such as in the factored generative models (Snyder et al., 2008, 2009; Naseem et al.,

Table 7.1. *The manually specified universal syntactic dependency rules of Naseem et al. (2010). Each rule (*head → dependent*) specifies which pairs of parts of speech are universally allowed to engage in a dependency relation.*

Root → Auxiliary	Noun → Adjective
Root → Verb	Noun → Article
Verb → Noun	Noun → Noun
Verb → Pronoun	Noun → Numeral
Verb → Adverb	Preposition → Noun
Verb → Verb	Adjective → Adverb
Auxiliary → Verb	

2009; Chen et al., 2011) and in models with partially shared parameters (Cohen and Smith, 2009; Berg-Kirkpatrick and Klein, 2010) for syntactic parsing, as well as in similar models for morphological segmentation (Snyder and Barzilay, 2008) and semantic role labeling (Titov and Klementiev, 2012). Others make this assumption explicit in the use of manually crafted universal rules, such as those shown in table 7.1, which were proposed for syntactic dependency parsing by Naseem et al. (2010). While these rules do not specify how to parse a particular sentence, they capture the most pertinent grammatical relations from a high level. By suitably constraining the parser to obey these constraints, either for every sentence or in expectation, the model can be biased towards sensible analyses. Kuhn (2004) proposed a different type of manually crafted rules, which specify how automatically induced word alignments (see below) restrict the syntactic constituency structure of different languages. Another example is the work of Schone and Jurafsky (2001), who use language universals, such as the fact that word classes exhibit similar frequency distributions across languages, to label automatically inferred word clusters with part-of-speech tags. See also the recent work of Zhang et al. (2012), who perform a similar experiment as part of their study of methods for automatically mapping fine-grained language specific part-of-speech tag sets onto a universal coarse-grained tag set. Note that the latter two are not strictly unsupervised approaches, as they require partial knowledge of part-of-speech tag characteristics.

A subset of languages supervised

In between the two previous scenarios, we have the scenario where supervision is available for a subset of the languages of interest. Since the seminal work by Yarowsky and Ngai (2001) and Yarowsky et al. (2001), this has been the most commonly studied setting. Assumptions similar to those discussed above are typically made in this setting as well. However, since the supervision is highly asymmetrical, these approaches typically focus on using the information available in the resource-rich language(s) to learn models for the

resource-poor language(s), rather than performing joint learning across both supervised and unsupervised languages.

This is the scenario considered in subsequent chapters and we return to a more detailed discussion of the two dominant approaches in this scenario in section 7.2. But first, let us motivate why we believe that this scenario is the most interesting one to study.

7.1.2 Arguments For Cross-Lingual Learning

The purely unsupervised scenario has given rise to many interesting structured models and algorithms. However, we believe that the focus on purely unsupervised methods is somewhat problematic for two reasons.

First, although these methods are often quite involved from a modeling perspective and require cutting-edge inference and learning techniques, in most studies they have not been shown to perform much better than quite impoverished unsupervised baselines, with performance far below simple supervised baselines trained with small amounts of labeled data. This is true even after considering the improvements from learning with multiple languages discussed above. For example, in many studies on unsupervised syntactic parsing these methods fare only marginally better than a naïve left/right-branching baseline. This is the case, for example, with the dependency model with valence (DMV) and constituent-context model (CCM) (Klein and Manning, 2002, 2004; Klein, 2005), which are commonly used as baselines in studies on unsupervised dependency and constituency parsing, respectively. Another severe restriction in many studies is that accuracy is typically measured only on sentences of ten or less words. When all sentence lengths are considered, a substantial drop in accuracy is often observed. This effect can be observed in the recent PASCAL shared task on unsupervised part-of-speech tagging and syntactic parsing (Gelling et al., 2012).[3] For the case of syntactic dependency parsing, the results show that the unlabeled attachment score of the best participating system drops from 63% on sentences of length 10 or less to 51% when considering all sentence lengths. The corresponding drop in accuracy for the best participating part-of-speech tagging system, when measured with an optimistic many-to-one mapping, is from 82% to 76%. Furthermore, as shown by Headden III et al. (2008), in the context of unsupervised part-of-speech tagging, there is little correlation between the performance of unsupervised methods, as measured by commonly used evaluation metrics, and the utility of the induced tags for down-stream tasks, such as syntactic parsing. On these grounds, the practical usefulness of purely unsupervised approaches to linguistic structure prediction can be questioned.

[3] See the updated version of Gelling et al. (2012), as table 3 in the original version of the paper conflated the results for different sentence lengths.

Second, the scenario where no supervision is available in any language may be unrealistic in practice. For example, for both part-of-speech tagging and syntactic parsing, which are the most commonly studied tasks in this context, fully annotated resources are available in a number of languages. The same holds for virtually any task studied by the natural language processing community, not least since the common evaluation protocol is based on comparison of predictions against manually created gold standards. We would even venture to argue that if a task, which in the end is to be measured against human accuracy, truly has practical importance, then someone will produce at least some labeled data in the world's major languages. Clearly, if our goal is to create models and methods that are practically useful, we should strive to take advantage of this rich source of linguistic information.

The scenario where supervision is available in all languages is also unlikely to occur for most, if not all, natural language processing tasks. For these reasons and because we believe it has the most practical potential, we subsequently focus solely on the scenario where supervision is available in a subset of languages. Furthermore, in order for our evaluations to be as realistic as possible, we include sentences of all lengths in our evaluations, with one exception: In the empirical study in chapter 9, we restrict ourselves to sentences of 50 words or less in order to facilitate comparison to closely related prior work that made this restriction. Since the average sentence length — at least in general English — is less than 25 words (Kummerfeld et al., 2012), this is a much less severe restriction than the common restriction to sentences of at most ten words.

7.2 Annotation Projection and Model Transfer

We next describe the two major approaches to cross-lingual learning with supervision available in a subset of languages: *annotation projection* (Yarowsky and Ngai, 2001; Yarowsky et al., 2001; Diab and Resnik, 2002; Hwa et al., 2005; Fossum and Abney, 2005; Padó and Lapata, 2006; Ganchev et al., 2009; Spreyer and Kuhn, 2009; Smith and Eisner, 2009; Das and Petrov, 2011) and *model transfer* (Zeman and Resnik, 2008; Øvrelid, 2009; McDonald et al., 2011; Søgaard, 2011; Cohen et al., 2011; Naseem et al., 2012; Søgaard and Wulff, 2012). In the former, the aim is to project annotations from resource-rich source language(s) to resource-poor target language(s), while in the latter, the aim is instead to directly transfer a model trained on the source language(s) to the target language(s).

Both of these approaches are employed in subsequent chapters as follows:

- In chapter 8, we introduce the idea of cross-lingual word clusters for model transfer; specifically, these clusters are evaluated for transfer of syntactic dependency parsing and named-entity recognition models.

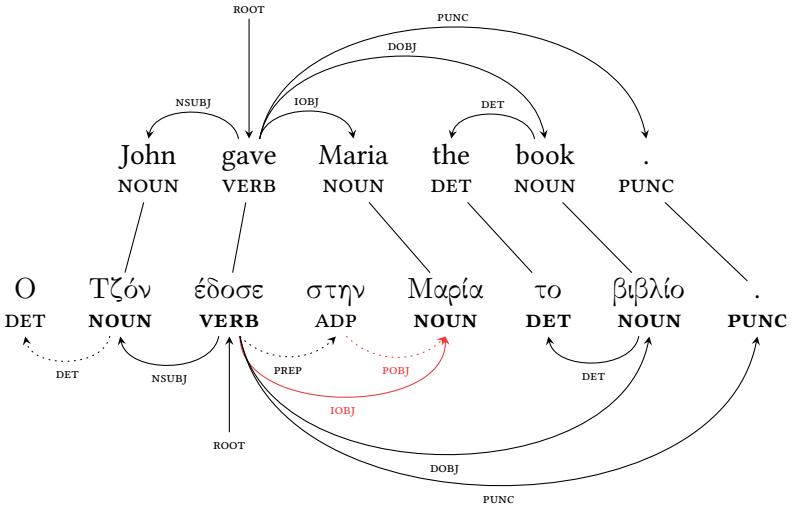

Figure 7.1. Projection of parts of speech and syntactic dependencies from English to Greek, based on the direct correspondence assumption (Hwa et al., 2005). The word alignments are derived from high-confidence intersected bidirectional alignments. Boldface part-of-speech tags indicate correctly projected tags. Remaining target language tags are left undetermined. Black solid arcs indicate correctly projected dependencies, while red solid arcs indicate incorrectly projected dependencies. Dotted arcs indicate gold dependencies that are left undetermined, while red dotted arcs indicate gold dependencies in which the head-attachment was erroneously projected.

- In chapter 9, we show how typological features can be used for selective parameter sharing in multi-source cross-lingual transfer (see section 7.2.3) of dependency parsing models.
- In chapter 10 we propose a method where annotation projection is utilized as part of a token and type constrained approach to cross-lingual part-of-speech tagging.

7.2.1 Annotation Projection

The most common approach to leverage linguistic annotations in a resource-rich language is by means of bitext which has been manually or automatically aligned, typically at the word level. Figure 7.1 shows an example of an English–Greek sentence pair which has been aligned in this way. The figure also shows how the part-of-speech and syntactic dependency annotations have been projected from the English side to the Greek side, via the word alignments. This way of projecting linguistic structure via bitext alignments was pioneered by Yarowsky and Ngai (2001) and Yarowsky et al. (2001), who leveraged word alignments to project predicted parts of speech, syntactic

chunks, named entities and morphological analyses from English to French, Chinese, Czech and Spanish.

Direct linguistic correspondence

The assumption that linguistic structure can be directly projected via word alignments in this way is often referred to as the *direct correspondence assumption*, a term introduced by Hwa et al. (2002, 2005). In essence, this assumption says that if a linguistic relation holds between two lexical items in the source language, then the same relation should hold between the corresponding lexical items in the target language translation, where the lexical correspondence is given by the alignments. Hwa et al. (2005) provide a formal statement of this assumption in the context of transfer of syntactic structure. However, the term has since often been used informally to refer to the general idea of naïvely projecting linguistic structure via word alignments. As discussed by several authors, there are a number of problems with this assumption (Bouma et al., 2008; Spreyer, 2011).

First, and foremost, translation is often non-literal, which means that there may not even exist any direct correspondence between the lexical items in a source text and its translation. Therefore, one may need to mine large quantities of translated text in order to find a sufficient amount of literal translations. While the automatic discovery and extraction of bitext suitable for lexical alignment and the inference of alignments in the extracted bitext is in itself a highly complex problem, this process is treated more or less as a black box in this dissertation. We refer the interested reader to Tiedemann (2011), who provides a comprehensive overview of the rich field of bitext alignment.

Second, even in literal translations, there is seldom a direct correspondence between all lexical items in the source and target languages. This is especially problematic in the case of function words, since these exhibit a high-degree of cross-lingual variability. There is a rich literature on structural and lexical variability. Spreyer (2011) provides a more detailed overview of this interesting topic. See also Stymne (2012), who discusses strategies for cross-lingual text harmonization in the context of machine translation, with a focus on compounding, definiteness and word order.

Figure 7.1 illustrates two instances of cross-lingual variability: in Greek, proper names in nominal subject position often take a determiner (Ο Τζόν), which is not the case in English (*John*); furthermore, in this example, the indirect object (*Maria*) in English corresponds to the prepositional phrase (στην Μαρία) in Greek. Due to these syntactic divergences, only parts of the linguistic structure is directly projected correctly from English to Greek (and vice versa). In this particular example, the projected part-of-speech annotation has a precision of 100% and a recall of 75% (6 correct and 2 undetermined tags), while the projected syntactic dependency annotation has a precision of 83% and a recall of 63% (5 correct, 1 incorrect and 2 undetermined dependencies). Note that in this example, the named entities *John* and *Maria* happen

to transfer perfectly between English and Greek; however, this cannot be expected in general, especially for named entities that span multiple tokens. In general, linguistic structure that span multiple tokens, such as dependency relations, phrase structure and, in some cases, named entities, can be expected to be less directly transferrable via word alignments, compared to token level structure, such as parts of speech. This is suggested by the difference in the score of the projected part-of-speech annotation and the projected dependency annotation in the above example. Note that the transfer example in fig. 7.1 is overly simplistic. In the study of Spreyer (2011) on a realistic corpus of English–German bitext, where syntactic dependencies projected from English are compared against gold standard annotation on the German side, the authors observe a precision of 67% and a recall of 36% (an F_1-score of 46%). Similarly, Hwa et al. (2005) report F_1-scores slightly below 40% for transfer of manually annotated syntactic dependencies from English to Spanish and Chinese, via manually created alignments, while Yarowsky and Ngai (2001) report a part-of-speech transfer accuracy of 85% using manually created alignments from English to French.[4]

Third, even in the case of literal translations with perfect lexical correspondence, noise may enter the transfer process via erroneous word alignments, when an automatic aligner is used, as well as via erroneous source side annotations, when these are automatically predicted. For example, Yarowsky and Ngai (2001) report a drop in part-of-speech tagging transfer accuracy from 85% to 76% when automatically induced word alignments are used in place of manually created alignments. We would expect structures involving multiple tokens to be more likely to be negatively impacted by both of these types of error. However, we are not aware of any study that has verified this hypothesis empirically.

Filtering, smoothing and relaxed correspondence

A range of different methods have been proposed for coping with the limitations of the direct correspondence assumption. These can be divided into those that work by correcting or filtering the incomplete and noisy annotations, those that apply smoothing techniques when training target side models on the projected annotation and those that relax the strong direct correspondence assumption.

In projecting syntactic dependencies from English to Spanish and Chinese, Hwa et al. (2005) showed that the accuracy of projected syntactic dependencies can be dramatically improved by employing a set of a dozen or so manually constructed post-projection transformation rules. As an example, one of the rules states that "The word preceding the token *di* should be labeled as an adverb, and modifies *di*, and *di* modifies the verb to its right." Despite

[4]It is unclear from Yarowsky and Ngai (2001) whether the reported accuracy corresponds to precision or recall.

their simplicity, these rules are quite effective. After application to the syntactic dependencies projected from English to Spanish and Chinese, they observe an improvement in F_1-scores from 37% to 70% and from 38% to 67%, respectively. Since the specification of such rules requires non-trivial linguistic competence in the target language — as well as manual inspection of the projected syntactic annotation — this approach may not scale so well to a truly multilingual scenario.

In addition to post-transformation rules, Hwa et al. (2005) apply a simple heuristic, which filters the automatically aligned bitext. The filter places a lower threshold on the fraction of source and target words that are required to be aligned, in order for a sentence to be used for annotation projection (between 70% to 80% of the tokens are required to be aligned, based on performance on a held-out development data set). Similar filtering strategies are evaluated by Spreyer (2011) for the transfer of syntactic dependencies from English to German. These filtering strategies drastically reduce the amount of bitext that can be used for annotation projection. Hwa et al. (2005) report that only about 20% of the aligned sentences remain after filtering, while only between 2% to 4% of the sentences remain after application of the most restrictive filter proposed by Spreyer (2011). Clearly, much information is lost when such aggressive filtering is employed. Spreyer (2011) therefore proposed to use a less restrictive filter, which leaves many of the projected dependency trees incompletely specified (an example of such incomplete transfer is illustrated in fig. 7.1). Instead, she introduced simple methods for training parsing models with incomplete trees. In contrast to the approach that we take in chapter 10 to cross-lingual part-of-speech tagging, where we marginalize over incompletely specified structure, her methods simply ignore any incomplete structure during training; they thereby lose the potential to leverage structural constraints and feature correlations for imputing the missing structure.

Filtering the bitext is a simple way to obtain word alignments of higher quality and thereby higher-quality annotation projection. However, as discussed above, even with perfect alignments, the projected annotation is likely to be noisy. Several methods for filtering the resulting annotation have therefore been proposed. One approach is to use an ensemble method, where multiple transfer sources are combined, with the hope that different transfer sources incur different errors, so that the errors may cancel out when the different sources are combined. In this vein, Yarowsky et al. (2001) used different translations of each sentence in the Bible for filtering annotation noise in cross-lingual transfer of morphology. Fossum and Abney (2005) similarly combined part-of-speech annotations projected from multiple source languages. Rather than using multiple translations of the same sentence for filtering, the latter aggregated information over the projected annotations; this aggregated information was then used used to filter the individual annotations. Similarly, Bouma et al. (2008) used multiple source languages for cor-

recting and filtering projected syntactic annotation. If the different sources have different biases, which is particularly likely when projecting from multiple languages, such filtering is able to filter out not only random noise, but also the systematic errors incurred when projecting from a single language. We return to the topic of multi-source cross-lingual learning in section 7.2.3.

Another filtering and smoothing approach was recently proposed by Das and Petrov (2011), who aggregate over projected part-of-speech annotation to create a type-level tag dictionary that restricts the set of potential parts of speech to which a word can be assigned. The dictionary is subsequently pruned with a simple thresholding heuristic, which removes low-confidence tags, after which the filtered tag dictionary is expanded to out-of-vocabulary words by means of label propagation (Zhu et al., 2003). A target language part-of-speech tagger is subsequently induced on unlabeled target text, treating the expanded dictionary as ambiguous supervision. In chapter 10, we propose another way to filter part-of-speech annotation projected from English to a range of other languages, by means of a crowd sourced tag dictionary. Our approach differs from those above in that we use the type-level dictionary to filter the token-level annotation, which like multi-source transfer is able to conquer both random and systematic noise; we then use both token-level and type-level supervision to train a target language part-of-speech tagger.

Noise can also be conquered by applying aggressive smoothing when fitting the parameters of the target language model, to prevent the model from overfitting to the noisy projected annotation (Yarowsky and Ngai, 2001; Moon and Baldridge, 2007). The method of Ganchev et al. (2009) can be seen as a way to relax the direct correspondence assumption and as a form of smoothing. They use projected incomplete syntactic dependency trees as soft constraints on posterior moments (Ganchev et al., 2010) when training a target language model; the use of soft in place of hard constraints reduces the risk of overfitting. Note that these smoothing and filtering approaches are unlikely to reduce noise resulting from systematic cross-lingual divergence and systematic alignment errors.

As pointed out by Hwa et al. (2005), the direct correspondence assumption also underlies synchronous grammars (Lewis and Stearns, 1968; Aho and Ullman, 1972), such as ITGs and 2-MTGs and hierarchical phrase-based models. Burkett et al. (2010) motivated their weakly synchronized joint parsing and alignment model as a way to relax the rigid direct correspondence assumption. Similar motivations were given by Eisner (2003) in proposing synchronous tree substitution grammars and by Smith and Eisner (2009) for their bilingual model based on quasi-synchronous grammar (Smith and Eisner, 2006). These synchronous and quasi-synchronous grammars could all theoretically be used for annotation projection, by inducing the joint generative model of source and target side syntax, while treating the source syntax as fixed and observed. However, to our knowledge, Smith and Eisner (2009) are the only ones to use this approach for annotation projection.

Finally, we note that there are also efforts to create parallel treebanks of two or more languages, where sentences in all included languages are both aligned and annotated with linguistic structure (Buch-Kromann et al., 2009; Ma, 2010; Adesam, 2012; Li et al., 2012; Haulrich, 2012).

7.2.2 Model Transfer

In contrast to annotation projection, most approaches to *model transfer* do not require the availability of aligned bitext (Zeman and Resnik, 2008; McDonald et al., 2011; Cohen et al., 2011; Søgaard, 2011). Instead, a model is trained on annotations in a source language, relying solely on features which are available in both the source and the target language. Typically, this involves removing all lexical features, a process referred to as *delexicalization*.[5]. Since all the features used by the model are also available in the target language, the model can be directly applied to the target language. At the time of writing, features that have been used in this way include "universal" coarse-grained part-of-speech tags (Zeman and Resnik, 2008; Naseem et al., 2010; Petrov et al., 2012), type-level lexical translation features (glosses) automatically extracted from word aligned bitext (Zeman and Resnik, 2008) and from bilingual dictionaries (Durrett et al., 2012), as well as cross-lingual distributed representations (Klementiev et al., 2012). In chapter 8, we introduce an additional feature type in the form of cross-lingual word clusters, that is, a grouping of words in two (or more languages), such that the groups are consistent across languages. Note that the cross-lingual distributed representations and the cross-lingual word clusters both rely on the availability of word aligned bitext. The above features are all word-level. In contrast, Øvrelid (2009) employs higher-level syntactic and morphological features for cross-lingual animacy classification.

There are three basic assumptions underlying model transfer approaches. First, that models for predicting linguistic structure, for example, syntactic dependencies, can be learned reliably using coarse-grained statistics, such as part-of-speech tags, in place of fine-grained statistics such as lexical word identities.[6] Second, that the parameters of features over coarse-grained statistics are in some sense language independent; in syntactic dependency parsing, for example, that a feature which indicates that adjectives modify their closest noun is useful in all languages. Third, that these coarse-grained statistics are robustly available across languages.

Considering the substantial differences between languages at the grammatical and lexical level, the prospect of directly applying a model trained on

[5]When transferring between very closely related languages, such as the Scandinavian languages, this may not be necessary (Skjærholt and Øvrelid, 2011)

[6]Note that the transferred model can be "relexicalized" by self-training on target language text with lexical features reinstated; see chapter 9.

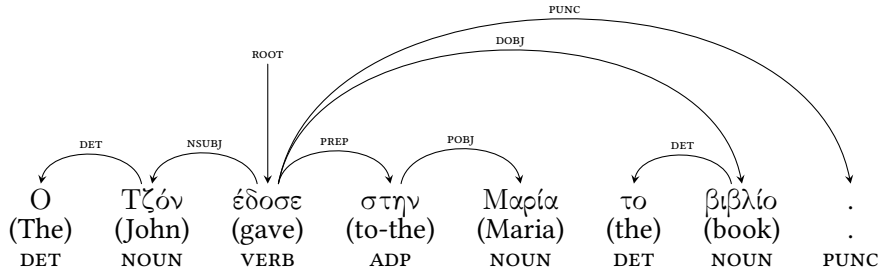

Figure 7.2. The same Greek sentence as in fig. 7.1, which is correctly parsed with a delexicalized dependency parser trained on annotated English text. Note that this assumes that the part-of-speech tags are available both in the source language and the target language.

one language to another language may seem bleak, even when such coarse-grained features are aligned across the languages. However, recent work has shown that a language independent syntactic dependency parser can indeed be created by training on a delexicalized treebank and by only incorporating features defined on coarse-grained "universal" part-of-speech tags (Zeman and Resnik, 2008; McDonald et al., 2011; Cohen et al., 2011; Søgaard, 2011). For languages with similar typology, this method can be quite accurate, especially when compared to purely unsupervised methods. To give an example, a syntactic parser trained on English with only coarse-grained part-of-speech tag features can correctly parse the Greek sentence in fig. 7.2, even without knowledge of the lexical items, since in this case the sequence of tags determines the syntactic structure almost unambiguously. Furthermore, as argued by McDonald et al. (2011), the selectional preferences learned by the model over part-of-speech tags is often sufficient to overcome minor differences in word order. Currently, the performance of even the simplest model transfer systems for syntactic dependency parsing far exceeds that of unsupervised systems for this task (Cohen et al., 2011; McDonald et al., 2011; Søgaard, 2011). Recall further that the direct correspondence assumption fails to properly transfer the correct annotation from the corresponding English translation in this case, as shown in fig. 7.1.

7.2.3 Multi-Source Transfer

Several studies have shown that combining multiple languages can benefit multilingual and cross-lingual learning (Cohen and Smith, 2009; Snyder and Barzilay, 2010; Berg-Kirkpatrick and Klein, 2010). We mentioned above how multiple source languages were used for filtering projected annotation by Fossum and Abney (2005) and Bouma et al. (2008). Annotations in multiple

languages can also be combined for model transfer, as long as the model features are available in all source languages as well as in the target language. This idea was first explored by McDonald et al. (2011), who showed that target language accuracy can be improved significantly by simply concatenating delexicalized treebanks in multiple languages when training a delexicalized transfer parser. Similar approaches were proposed independently by Cohen et al. (2011) and Søgaard (2011). The former uses a mixture model in which the parameters of a generative target language parser are expressed as a linear interpolation of source language parameters, where the interpolation weights are estimated using expectation-maximization on unlabeled target data. and. Søgaard (2011) instead uses an n-gram language model trained on the target language to selectively subsample training sentences from the source languages' training data. The idea is that the perplexity of the source sentences according to the target language model can be used to sample source sentences that are more similar to the target language. Recently, Søgaard and Wulff (2012) proposed additional methods for source language instance weighting.

These multi-source model transfer methods assume that the target language can be expressed as a linear combination of source languages. As we discuss in chapter 9, this assumption is not realistic, as different languages tend to share varied typological traits. Inspired by another recent method of Naseem et al. (2012), we introduce a method which makes use of typological information to selectively share parameters, such that the target language model shares different features with different source languages. The typological information is extracted from a publicly available database (Dryer and Haspelmath, 2011).

7.3 Cross-Lingual Evaluation

As discussed, cross-lingual divergence can be both a blessing (by enabling ambiguity resolution, in particular in the multi-source setting) and a curse (by introducing errors and incomplete annotation projection). These divergences notwithstanding, there may be multiple ways of analyzing the same linguistic construction, such that there is no consensus as to which analysis is preferable. Different treebanks therefore typically have more or less diverging annotation guidelines. Naturally, treebanks for different languages typically disagree to a larger extent compared to multiple treebanks in the same language. However, even in the same language differences in annotation guidelines between treebanks can be an issue (Smith and Eisner, 2009). For example, different treebanks often define differing part-of-speech tag sets, both in terms of tag granularity and in terms of broader tag distinctions. Another example is annotation of syntactic dependencies, where some syntactic constructions, such as coordination (Nivre, 2006; Maier et al., 2012), are no-

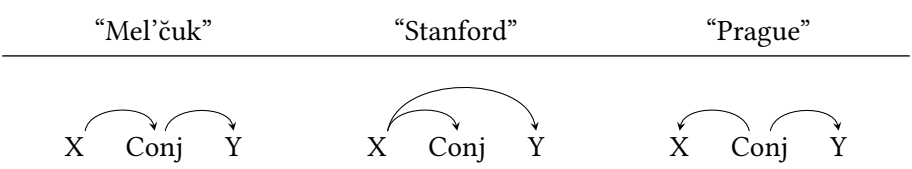

Figure 7.3. Treatment of coordinating conjunctions in different syntactic dependency annotation schemes. Figure inspired by Spreyer (2010).

toriously afforded competing analyses in different treebanks. The latter is illustrated in fig. 7.3, which shows three different ways of annotating coordination: according to (left) "Mel'čuk"-style rules (Nilsson et al., 2007); (center) "Stanford"-style conversion rules (de Marneffe and Manning, 2008); and (right) "Prague"-style rules (Böhmová et al., 2003). Spreyer (2010), from which fig. 7.3 is adapted, analyzes which syntactic constructions differ most across treebanks in English, German, Dutch and Italian. In addition to coordination, she finds that the most diverging constructions are prepositional phrases with embedded noun phrases, auxiliary verb constructions, subordinating clauses and relative clauses. Since these are all common constructions, one can expect substantial cross-lingual divergence resulting from incompatible annotation guidelines. This is a problem in particular for the evaluation of transferred models or annotations against a target language gold standard. Preferably, divergence that is caused by more or less arbitrary annotation guideline decisions, should be separated from true linguistic divergence. This issue can be addressed in two different ways: either by harmonizing the annotation of the source and the target language, or by relaxing the evaluation measure, so that it becomes agnostic to treebank-specific annotation choices.

The approach of Ganchev et al. (2009) belongs to the former category. They use a small number of rules to harmonize syntactic dependency annotation transferred from English to Bulgarian and Spanish, respectively. They observe a relative error reduction of more than 30% for Bulgarian and 5% for Spanish, compared to using the raw projected annotation, when evaluating against the target language gold standard. The dramatic impact on Bulgarian is attributable to the harmonization of auxiliary verb constructions. The rules seem to have been designed partly by considering differences in annotation guidelines and in analyzing transfer errors. Another example is Hwa et al. (2005), who use post-transformation rules, which partly correct the projected annotation. These rules do not seem to be based directly on treebank divergence considerations, but rather on an analysis of transfer errors. A more linguistically principled attempt to harmonize the syntactic annotations of treebanks in multiple languages is currently being undertaken by Zeman et al. (2012). While incomplete at the time of writing, this effort should become a highly useful resource in cross-lingual learning research upon completion.

In the context of cross-lingual part-of-speech tagging, more progress has been made in this regard, with several different "universal" tag sets available (Zeman and Resnik, 2008; Naseem et al., 2010; Petrov et al., 2012). These specify a dozen or so coarse-grained tags available across languages, together with mappings from more fine-grained language specific tag sets to the set of coarse-grained tags. While these mappings have been manually specified, based on inspection of annotation guidelines, Zhang et al. (2012) propose an automatic way of mapping fine-grained tags to a common coarse-grained representation.

In addition to these harmonization and post-transfer error correction attempts, there are two recent related proposals on using relaxed evaluation measures to mitigate the issues of cross-annotation evaluation of syntactic dependency parsers. Schwartz et al. (2011) suggest a simple evaluation measure insensitive to direction, that is, a measure which is agnostic to the designation of head and dependent in a dependency relation. Tsarfaty et al. (2011) go further and develop a more general framework for evaluation across linguistic formalisms. They propose a family of evaluation measures based on tree transformations coupled with a cost function formulated in terms of tree edit distance. Both of these approaches allow downplaying differences in, for example, phrase-internal structure, such as coordination, where annotation decisions are more or less arbitrary.

While these evaluation methods may certainly have their use, there are considerations to be made when switching to such relaxed measures in place of the rigid evaluation measures currently used, since it is unclear exactly how much information is lost in the process. Instead, we believe that it would be preferable to define harmonized annotation guidelines for as many languages as possible, since this would make it more clear exactly what is lost when transferring between different languages. Unfortunately, such a harmonization is not yet available, although Zeman et al. (2012) is an encouraging effort towards this goal. Rather than following previous work, such as Hwa et al. (2005) and Ganchev et al. (2009), when studying cross-lingual transfer methods for dependency parsing, we only evaluate against the existing treebank annotations. The cross-lingual results presented in subsequent chapters are therefore likely to be somewhat underestimating the actual performance of the transfer systems. This hypothesis is partly confirmed in chapter 8, where we perform an additional evaluation against a small German treebank, which has been annotated according to the same guidelines as the English source parser. This comparison shows that evaluating against the native German annotation substantially underestimates the transfer parser's true ability to predict syntactic dependencies.

When studying cross-lingual part-of-speech tagging in chapter 10, we evaluate against the coarse-grained universal tag set of Petrov et al. (2012). Disregarding the issue of potential errors in the mappings from fine-grained

treebanks specific tag sets to the universal coarse-grained tags, these results should not suffer from underestimation.

8. Cross-Lingual Word Clusters for Model Transfer

The previous chapter provided an overview of different approaches to multilingual and, in particular, cross-lingual prediction of linguistic structure. Before this, chapter 5 discussed methods for structured prediction with incomplete or no supervision. This chapter integrates these settings by leveraging unsupervised feature induction and word-aligned bitext for improved cross-lingual model transfer.

It has been established that incorporating features based on word clusters induced from large unlabeled text corpora can significantly improve prediction of linguistic structure. Previous studies on such features for semi-independent learning have typically focused only on a small set of languages and tasks. The first half of this chapter describes how monolingual word clusters can be induced with an unsupervised class-based language model, and how the induced clusters can be used as features for semi-supervised learning. Empirically, these features are shown to be robust across 13 languages for dependency parsing and across 4 languages for named-entity recognition. This is the first study with such a broad view on this subject, in terms of language diversity. As we will see in chapter 10, these word clusters are also highly useful for learning part-of-speech taggers with ambiguous supervision in a multilingual setting.

Encouraged by the results in the monolingual learning setting, we turn to the cross-lingual transfer setting. Recall the delexicalized transfer approach from section 7.2.2, where features derived solely from "universal" part-of-speech tags are typically employed. These features are often too coarse-grained to capture many linguistic phenomena, such as selectional restrictions; see section 2.3. In the second half of this chapter, we consider the use of word-cluster features as a way to address the problem of feature granularity in this scenario. To this end, we develop an algorithm which generates cross-lingual word clusters; that is, clusters of words that are consistent across (pairs of) languages. This is achieved by coupling two instances of the class-based language model used in the monolingual setting — one for each language — via word-aligned bitext through which cross-lingual word-cluster constraints are enforced. We show that the inclusion of features derived from these cross-lingual clusters significantly improves the accuracy of cross-lingual model transfer of syntactic dependency parsers and named-entity recognizers.

8.1 Monolingual Word Clusters

Word-cluster features have been shown to be useful in various tasks in natural language processing, including syntactic dependency parsing (Koo et al., 2008; Haffari et al., 2011; Tratz and Hovy, 2011), syntactic chunking (Turian et al., 2010), and named-entity recognition (Freitag, 2004; Miller et al., 2004; Turian et al., 2010; Faruqui and Padó, 2010). The effectiveness of word-cluster features derives from their ability to aggregate local distributional information from large unlabeled corpora, which aids in conquering data sparsity in supervised training regimes as well as in mitigating cross-domain generalization issues (provided that the unlabeled data span multiple domains). Hitherto, most studies on the use of word-cluster features in NLP have focused only on English, though there have been exceptions, such as Faruqui and Padó (2010) who looked at German named-entity recognition and Koo et al. (2008) who studied Czech dependency parsing. In the next sections, we address this issue by examining the versatility of word clusters for dependency parsing and named-entity recognition across a range of languages. We will see that such clusters are indeed almost universally useful. This is encouraging for the second half of the chapter, in which we adapt the model described in this section to the cross-lingual setting.

In line with much previous work on word clusters for tasks such as dependency parsing and named-entity recognition, for which local syntactic and semantic constraints are of importance, we induce word clusters by means of an unsupervised class-based language model. Models of this type employ latent variables and Markov assumptions to effectively model sequences of words. By constraining the latent variables to a single state for each word type, a hard clustering over the words is achieved. Rather than the more commonly used model of Brown et al. (1992), we employ the predictive class bigram model introduced by Uszkoreit and Brants (2008). The two models are very similar, but whereas the former takes class-to-class transitions into account, the latter directly models word-to-class transitions. By ignoring class-to-class transitions, an approximate maximum-likelihood clustering can be computed efficiently. This is a useful property, as we later develop an algorithm for inducing cross-lingual word clusters that calls this monolingual algorithm as a subroutine and because we will cluster datasets with billions of tokens. While the use of class-to-class transitions can lead to more compact models, which is often useful for conquering data sparsity, when clustering large data sets we can get reliable statistics directly on the word-to-class transitions (Uszkoreit and Brants, 2008).

More formally, let $Z : \mathcal{V} \mapsto [1, K]$ denote a (hard) latent clustering function that maps each word type from the vocabulary $\mathcal{V}$ to one of K cluster identifiers. Let START be a designated start-of-segment symbol and let $x = (x_1, x_2, \ldots, |x|) \in \mathcal{X}$ be a sequence of word tokens, with $x_i \in \mathcal{V} \cup \{\text{START}\}$. With the model of Uszkoreit and Brants (2008), the joint probability of a se-

quence of tokens and a cluster assignment of these tokens factorizes as

$$p(x, Z) = \prod_{i=1}^{|x|} p(x_i \mid Z(x_i)) p(Z(x_i) \mid x_{i-1}) . \tag{8.1}$$

When the cluster assignment is unknown, this is a special case of the structured generative latent variable models described in section 5.2.2, in which the latent variables have been constrained to form a hard type-level clustering and where instead of marginalizing over the latent variables, we consider their MAP assignment. Let $\mathcal{Z}(x, \mathcal{V})$ be the set of latent variable assignments which satisfy the hard clustering constraint that $Z(x_i) = Z(v)$ for all $i \in [1, |x|]$ and $v \in \mathcal{V}$ such that $x_i = v$. Treating all the data as a long token sequence, the optimal clustering is that which minimizes the negative log-likelihood of this constrained model:

$$\widehat{Z} = \operatorname*{arg\,min}_{Z \in \mathcal{Z}(x, \mathcal{V})} J(x; Z) = \operatorname*{arg\,min}_{Z \in \mathcal{Z}(x, \mathcal{V})} \left[-\log p(x, Z) \right] . \tag{8.2}$$

Uszkoreit and Brants (2008) showed that one can optimize this objective function with a highly efficient distributed algorithm.

The difference between this model and that of Brown et al. (1992) becomes more clear when considering the model structure of the latter:

$$p(x, Z) = \prod_{i=1}^{|x|} p(x_i \mid Z(x_i)) p(Z(x_i) \mid Z(x_{i-1})) .$$

Whereas the former conditions the cluster assignment of the current token on the previous token, the latter conditions the cluster assignment on the cluster assignment of the previous token. This cluster-cluster coupling, makes the latter more difficult to optimize efficiently.

In addition to these class-based clustering models, a number of additional word clustering and embedding variants have been proposed (Schütze, 1993, 1995; Globerson et al., 2007; Maron et al., 2010; Lamar et al., 2010). The use of such models for semi-supervised learning has recently been explored. For example, Turian et al. (2010) assessed the effectiveness of the word embedding techniques of Mnih and Hinton (2007) and Collobert and Weston (2008) along with the word-clustering method of Brown et al. (1992) for syntactic chunking and named-entity recognition. Dhillon et al. (2011) proposed a word embedding method based on canonical correlation analysis that obtains state-of-the art results for word-based semi-supervised learning for English named-entity recognition; see also the recent follow-up paper by Dhillon et al. (2012). Another recent approach is the marginalized denoising autoencoders introduced by Chen et al. (2012), who report highly encouraging results for domain adaptation for sentiment analysis. Finally, as an alternative to the above word-based methods, Lin and Wu (2009) proposed a phrase-clustering method that currently attains the best published result for English named-entity recognition.

8.2 Monolingual Experiments

Before moving on to the multilingual setting, we conduct a set of monolingual experiments where we evaluate the use of the monolingual word clusters described in the previous section as features for syntactic dependency parsing and named-entity recognition.

8.2.1 Experimental Setup

Below we detail the experimental setup used in the monolingual experiments. These languages and data sets are also used for the cross-lingual experiments in section 8.4.

Languages and treebanks for dependency parsing

In the parsing experiments, we study the following thirteen languages: Danish (da), German (de), Greek (el), English (en), Spanish (es), French (fr), Italian (it), Korean (ko), Dutch (nl), Portugese (pt), Russian (ru), Swedish (sv) and Chinese (zh). These languages represent the Chinese, Germanic, Hellenic, Romance, Slavic, Altaic and Korean genera.[1] The two-letter abbreviations from the ISO 639-1 standard (in parentheses; see appendix A) are used when referring to these languages in tables and figures.

The following treebanks are used for these experiments. For Danish, German, Greek, Spanish, Italian, Dutch, Portuguese and Swedish, we use the predefined training and evaluation data sets from the CoNLL 2006/2007 data sets (Buchholz and Marsi, 2006; Nivre et al., 2007). For English we use sections 02-21, 22, and 23 of the Penn WSJ Treebank (Marcus et al., 1993) for training, development and evaluation. For French we used the French Treebank (Abeillé and Barrier, 2004) with splits defined by Candito et al. (2010). For Korean we use the Sejong Korean Treebank (Han et al., 2002) randomly splitting the data into 80% training, 10% development and 10% evaluation. For Russian we use the SynTagRus Treebank (Boguslavsky et al., 2000; Apresjan et al., 2006) randomly splitting the data into 80% training, 10% development and 10% evaluation. For Chinese we use the Penn Chinese Treebank v6 (Xue et al., 2005) using the proposed data splits from the documentation. Both English and Chinese are converted to dependencies using v1.6.8 of the Stanford Converter (De Marneffe et al., 2006). French is converted using the procedure defined in Candito et al. (2010). Russian and Korean are native dependency treebanks. For the CoNLL data sets we use the part-of-speech tags provided with the data. For all other data sets, we train a part-of-speech tagger on the training data and automatically tag the development and evaluation data.

[1] The particular choice of languages for this empirical study was based purely on data availability and institution licensing.

Languages and data sets for named-entity recognition
In the named-entity recognition experiments, we study German, English and Dutch (three Germanic languages) and Spanish (a Romance language). For all four languages, we use the training, development and evaluation data sets from the CoNLL 2002/2003 shared tasks (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003). The data set for each language consists of newswire text annotated with four entity categories: Location (LOC), Miscellaneous (MISC), Organization (ORG) and Person (PER). We use the part-of-speech tags supplied with the data, except for Spanish where we instead use universal part-of-speech tags (Petrov et al., 2012), since the Spanish CoNLL data lacks part-of-speech annotation.

Unlabeled data for clustering
The unlabeled data for inducing the monolingual clusters is extracted from one year of newswire articles from multiple sources from a news aggregation website. This data consists of between 800 million (Danish) and 121.6 billion (English) tokens per language.

For all experiments we fix the number of clusters to 256 as this performed best on held-out data. While combining different cluster granularities has been shown to be useful (Turian et al., 2010; Razavi, 2012), for reasons of simplicity, we only consider one level of granularity in these experiments. Furthermore, we only cluster the one million most frequent word types in each language for reasons of efficiency and in order to conquer data sparsity. However, even when not all word types are clustered, all word types are included as context in the bigram statistics used for estimating the clustering model in eq. (8.1).[2] For languages where the unlabeled data does not contain at least one million word types, all word types are included in the clustering.

Dependency parsing model
All of the parsing experiments reported in this empirical evaluation are based on the transition-based dependency parsing paradigm (Nivre, 2008). For all languages and settings, we use an arc-eager decoding strategy with beam-search inference, using a beam of eight hypotheses, and we train the model for ten epochs with the averaged structured perceptron algorithm (Zhang and Clark, 2008). The parser is treated more or less as a black box in this chapter, although we do modify its feature model in the next section to incorporate word clusters. We refer the reader to Nivre (2008) for a detailed description of transition-based parsing and we note that a post-study experiment with an arc-factored graph-based parser on the same data sets suggest that the two types of parsing models give comparable results.

[2]This is another virtue of the model of Uszkoreit and Brants (2008), compared to the model of Brown et al. (1992), as the latter requires that all word types that are used as contexts in the clustering model are also clustered.

Table 8.1. *Additional cluster-based parser features. These features are added to the ones proposed by Zhang and Nivre (2011), whose notation we adapt and extend with clusters here. S_i and N_i: the* ith *tokens in the stack and buffer (indexing starts at 0). p: the part-of-speech tag of the token. c: the cluster of the token. v_l: the valence of the left set of children. v_r: the valence of the right set of children. l: the dependency label of the (dependent) token under consideration. d: distance between the words on the top of the stack and buffer. S_0h, S_0r and S_0l: the head, right-most dependent and left-most dependent of the token at the top of the stack. S_0h2, S_0l2, S_0r2: the head of S_0h, the second leftmost dependent and the second rightmost dependent of S_0. S_0l2: the second leftmost dependent of N_0.*

Single words	S_0c, S_0cp, N_0c, N_0cp, N_1cp, N_1c, N_2cp, N_2c
Word pairs	S_0cpN_0cp, S_0cN_0cp, S_0cpN_0c, S_0cN_0c, S_0pcN_0p S_0pN_0pc, S_0wN_0c, S_0cN_0w, N_0cN_1c, N_1cN_2c
Word triples	$N_0cN_1cN_2c$, $S_0cN_0cN_1c$, $S_{0h}cS_0cN_0c$, $S_0cS_{0l}cN_0c$, $S_0cS_{0r}cN_0c$, $S_0cN_0cN_{0l}c$
Distance	S_0cd, N_0cd, S_0cN_0cd
Valency	S_0cv_l, S_0cv_r, $N_0cS_0v_l$
Unigrams	$S_{0h}c$, $S_{0l}c$, $S_{0r}c$, $N_{0l}c$
Third-order	$S_{0h2}c$, $S_{0l2}c$, $S_{0r2}c$, $N_{0l2}c$
Labels	$S_0cS_{0l}l$, $S_0cS_{0r}l$, $N_0cN_{0l}l$, $N_0cN_{0r}l$

Named-entity recognition model

For all named-entity recognition experiments, we use a first-order linear-chain conditional random field; see section 3.4.1. For training the model, we use the log loss together with an ℓ_2-regularizer, setting $\lambda = 1$; see section 4.1.1 and section 4.1.4. Optimization is performed with the gradient-based L-BFGS algorithm (see section 4.2.1) until ϵ-convergence, with $\epsilon = 0.0001$. Convergence is typically reached after less than 400 iterations.

8.2.2 Cluster-Augmented Feature Models

We extend the state-of-the-art feature model introduced by Zhang and Nivre (2011) by adding an additional word cluster based feature template for each word based template. Additionally, we add templates where one or more part-of-speech feature is replaced with the corresponding cluster feature. The resulting set of additional feature templates are shown in table 8.1. The expanded feature model includes all of the feature templates defined by Zhang and Nivre (2011), which we also use as the baseline model, whereas table 8.1 only shows our new templates.

The feature model used for the named-entity tagger is shown in table 8.2. These are similar to the features used by Turian et al. (2010), with the main difference that we do not use any long range features and that we add templates that conjoin adjacent clusters and adjacent tags as well as templates that conjoin label transitions with tags, clusters and capitalization features.

Table 8.2. *Cluster-augmented named-entity recognition features. Hyp: Word contains hyphen. Cap: First letter is capitalized. Trans $\otimes$ f: First-order label transition from previous to current label, conjoined with feature f. $x^{:j}$: j-character prefix of x. $x^{-j:}$: j-character suffix of x. f_i: Feature f at relative position i. $f_{i,j}$: Union of features at positions i and j. $f_{i:j}$: Conjoined feature sequence between relative positions i and j (inclusive). Bias: Constant bias features.*

Word & bias	$x_{-1,0,1}$, $x_{-1:0}$, $x_{0:1}$, $x_{-1:1}$, Bias
Prefix	$x^{:1}_{-1,0,1}$, $x^{:2}_{-1,0,1}$, $x^{:3}_{-1,0,1}$, $x^{:4}_{-1,0,1}$, $x^{:5}_{-1,0,1}$
Suffix	$x^{-1:}_{-1,0,1}$, $x^{-2:}_{-1,0,1}$, $x^{-3:}_{-1,0,1}$, $x^{-4:}_{-1,0,1}$, $x^{-5:}_{-1,0,1}$
Orthography	$\text{Hyp}_{-1,0,1}$, $\text{Cap}_{-1,0,1}$, $\text{Cap}_{-1:0}$, $\text{Cap}_{0:1}$, $\text{Cap}_{-1:1}$
Part of speech	$p_{-1,0,1}$, $p_{-1:0}$, $p_{0:1}$, $p_{-1:1}$, $p_{-2:1}$, $p_{-1:2}$
Cluster	$c_{-1,0,1}$, $c_{-1:0}$, $c_{0:1}$, $c_{-1:1}$, $c_{-2:1}$, $c_{-1:2}$
Transition	$\text{Trans} \otimes p_{-1,0,1}$, $\text{Trans} \otimes c_{-1,0,1}$, $\text{Trans} \otimes \text{Cap}_{-1,0,1}$, Trans $\otimes$ b

8.2.3 Results

The results of the parsing experiments, measured with labeled attachment score (LAS, see section 4.1.2) on all sentence lengths, excluding punctuation, are shown in table 8.3. The baselines are all comparable to the state of the art. On average, the addition of word-cluster features results in a 6% relative reduction in error, with a relative reduction upwards of 15% for Russian. All languages improve, except French, which sees neither an increase nor a decrease in LAS. We observe an average absolute increase in LAS of approximately 1%, which is in line with previous observations by, for example, Koo et al. (2008). It is not surprising that Russian sees a large gain as it is a highly inflected language, making observations of lexical features far more sparse. Some languages, for example, French, Dutch and Chinese see much smaller gains. One likely culprit is a divergence between the tokenization schemes used in the treebank and in our unlabeled data, which for Indo-European languages is closely related to the Penn Treebank tokenization. For example, the Dutch treebank contains many multi-word tokens that are typically broken apart by our automatic tokenizer.

The results for named-entity recognition, in terms of the F_1 measure (see section 4.1.2), are listed in table 8.4. Introducing word-cluster features for named-entity recognition reduces relative errors on the test set by 21% (39% on the development set) on average. Broken down per language, reductions on the test set vary from substantial for Dutch (30%) and English (26%), down to more modest for German (17%) and Spanish (12%). The addition of cluster features most markedly improves recognition of the PER category, with an average error reduction on the test set of 44%, while the reductions for ORG (19%), LOC (17%) and MISC (10%) are more modest, but still significant. Although our results are below the best reported results for English and German (Lin and Wu, 2009; Faruqui and Padó, 2010), the relative improvements of adding word clusters are in line with previous results on named-entity

Table 8.3. *Results of supervised parsing, as measured with labeled attachment score (LAS) on the test set. All improvements are statistically significant ($p < 0.05$), except for French (fr) and Dutch (nl).*

Language	NO CLUSTERS	CLUSTERS
da	84.3	**85.8**
de	88.9	**89.5**
el	76.1	**77.3**
en	90.3	**90.7**
es	82.8	**83.6**
fr	85.7	85.7
it	81.4	**82.2**
ko	82.0	**83.6**
nl	77.2	**77.8**
pt	86.9	**87.6**
ru	83.5	**86.0**
sv	84.7	**86.5**
zh	74.9	**75.5**
avg	83.0	**84.0**

recognition for English. Miller et al. (2004); Turian et al. (2010) report error reductions of approximately 25% by adding word-cluster features. Slightly higher reductions where achieved for German by Faruqui and Padó (2010), who report a reduction of 22%. Note that we did not tune hyper-parameters of the supervised learning methods and of the clustering method, such as the number of clusters (Turian et al., 2010; Faruqui and Padó, 2010), and that we did not apply any heuristic for data cleaning such as that used by Turian et al. (2010).

8.3 Cross-Lingual Word Clusters

All results of the previous section rely on the availability of large quantities of language specific annotations for each task. Cross-lingual transfer methods have recently been shown to hold promise as a way to at least partially sidestep the demand for labeled data; see chapter 7. The aim of these methods is to use knowledge induced from labeled resources in one or more resource-rich source languages to construct systems for resource-poor target languages in which no or few such resources are available.

As discussed in chapter 7, an effective way to achieve this — in particular in the case of syntactic dependency parsing — is by means of delexicalized model transfer (Zeman and Resnik, 2008; McDonald et al., 2011). The approach works as follows: for a given training set, the learner ignores all lexical identities and only observes features over other characteristics, such as

Table 8.4. *Results of supervised named-entity recognition, as measured with F_1-score on the CoNLL 2002/2003 development and test sets.*

	de	en	es	nl	avg
NO CLUSTERS	65.4	89.2	75.0	75.7	76.3
CLUSTERS	**74.8**	**91.8**	**81.1**	**84.2**	**83.0**
↑ DEVELOPMENT SET ↓ TEST SET					
NO CLUSTERS	69.1	83.5	78.9	79.6	77.8
CLUSTERS	**74.4**	**87.8**	**82.0**	**85.7**	**82.5**

"universal" part-of-speech tags and direction of syntactic attachment. Since the model is trained strictly with features that are available across languages, a model trained on an annotated source language data set can directly be applied to target language text. This simple approach has been shown to outperform a number of state-of-the-art unsupervised and transfer-based baselines. However, the strict reliance on coarse-grained non-lexical features limits its effectiveness.

In the remainder of this chapter, we seek to address this issue by supplementing the restricted set of coarse-grained features with features derived from cross-lingual word clusters. A cross-lingual word clustering is a clustering of words in two languages, such that the clusters correspond to meaningful cross-lingual properties. For example, prepositions from both languages should be in the same cluster, proper names from both languages in another cluster, and so on. By adding features defined over these clusters, we can to some degree robustly re-lexicalize the delexicalized models, while maintaining the "universality" of the features. Figure 8.1 outlines our approach as applied to syntactic dependency parsing. Assuming that we have an algorithm for generating cross-lingual word clusters (see section 8.3), we can augment the delexicalized parsing algorithm to use these word-cluster features at training and testing time.

In order to further motivate the proposed approach, consider the accuracy of the supervised English parser. A parser with lexical, part-of-speech and cluster features achieves a LAS of 90.7; see table 8.3. If we remove all lexical and cluster features, the same parser achieves 83.1%. However, if we add back just the cluster features, the accuracy jumps back up to 89.5%, which is only 1.2% below the fully lexicalized parser. Thus, if we can accurately learn cross-lingual clusters, there is hope of regaining some of the accuracy lost due to the delexicalization process. We hypothesize that this is especially important for recovering dependency label information. While unlabeled dependencies are largely determined by part-of-speech information, successful prediction of relations such as SUBJ and OBJ is likely to require more fine-grained information. This is likely an issue particularly in the cross-lingual transfer

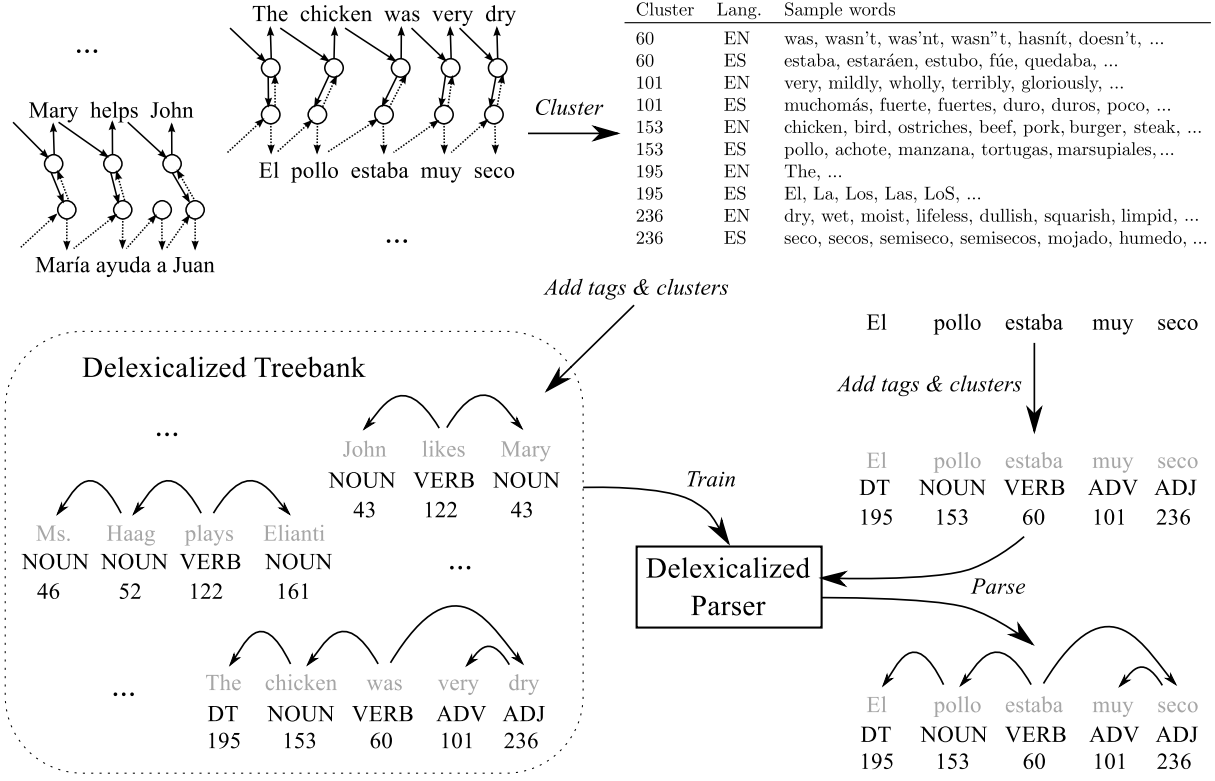

Figure 8.1. Illustration of the use of cross-lingual word clusters for cross-lingual transfer of a syntactic dependency parsing model. Top-left: Cross-lingual (English–Spanish) word clustering. Top-right: Sample of the cross-lingual word clusters. Bottom-left: Delexicalized cluster-augmented source (English) treebank for parser training. Bottom-right: Parsing of a target (Spanish) sentence with the transfer parser.

setting, where the parser is not able to rely on word-order information to the same degree.

8.3.1 Cluster Projection

Our first method for inducing cross-lingual clusters has two stages. First, the word types of a source language (S) is clustered as in the monolingual case; see section 8.1. We then project these monolingual source language clusters to a target language (T), using word-aligned bitext. Assume that we are given two aligned token sequences $x^S = (x_1^S, x_2^S, \ldots, x_m^S)$ and $x^T = (x_1^T, x_2^T, \ldots, x_n^T)$, where $m = \left|x^S\right|$ and $n = \left|x^T\right|$. Let $\mathcal{A}^{T|S}$ be a set of scored and filtered word-alignments from the source language sequence to the target language sequence, where $(x_i^T, x_{a_i}^S, s_{i,a_i}) \in \mathcal{A}^{T|S}$ is an alignment from the a_ith source token to the ith target token, with alignment score $s_{i,a_i} \geq \delta$. Here, the alignment score corresponds to the conditional alignment probability $p(x_i^T \mid x_{a_i}^S)$, provided by the alignment model. Throughout, we ignore all ϵ-alignments — that is, all unaligned tokens are ignored — and we use $\delta = 0.95$. The shorthand $i \in \mathcal{A}^{T|S}$ is used to denote the target tokens x_i^T that are aligned to some source token $x_{a_i}^S$.

Provided a monolingual source language clustering Z^S, the target word type $v^T \in \mathcal{V}^T$ is simply assigned to the cluster with which it is most often aligned, taking alignment scores into account:

$$Z^T(v^T) = \arg\max_{k \in [1,K]} \sum_{i \in \mathcal{A}^{T|S}} \mathbb{1}\left[x_i^T = v^T \wedge Z^S(x_{a_i}^S) = k\right] s_{i,a_i}\,, \tag{8.3}$$

where we treat the whole bitext as one long sequence of aligned tokens, so that $\mathcal{A}^{T|S}$ is the set of all alignments in the full bitext. We refer to the cross-lingual clusters assembled in this way as PROJECTED CLUSTERS.

8.3.2 Joint Cross-Lingual Clustering

The simple projection approach just described has two potential drawbacks. First, it only provides a clustering of those target language words that occur in the word-aligned bitext, which is typically orders of magnitude smaller than our monolingual data sets. Second, the mapped clustering may not necessarily correspond to an acceptable target language clustering in terms of monolingual likelihood. In order to address these issues, we propose the following more complex model. First, to find clusterings that are good according to both the source and target language, and to make use of more unlabeled data, we model token sequences in each language by the monolingual language model in eq. (8.1). Denote the objective functions — corresponding to eq. (8.2) — of these monolingual models by $J_S(x^S; Z^S)$ and $J_T(x^T; Z^T)$, respectively. Second, we constrain the clusterings defined by these individual models to agree

Algorithm 4 Cross-Lingual Word Clustering

Randomly initialize source and target clusterings Z^S and Z^T.
for $n = 1 \ldots N$ **do**
1. Find $\widehat{Z}^S \approx \arg\min_{Z^S \in \mathcal{Z}(x^S, \mathcal{V}^S)} J_S(x^S; Z^S)$. (†)
2. Project $\widehat{Z}^S$ to Z^T using eq. (8.3).
   — keep cluster of non-projected words in Z^T fixed.
3. Find $\widehat{Z}^T \approx \arg\min_{Z^T \in \mathcal{Z}(x^T, \mathcal{V}^T)} J_T(x^T; Z^T)$. (†)
4. Project $\widehat{Z}^T$ to Z^S using eq. (8.3).
   — keep cluster of non-projected words in Z^S fixed.

end for
† Equation (8.2) optimized with the exchange algorithm, keeping the cluster assignment from the projection step fixed.

via the word alignments. This is achieved with the simple alternating procedure outlined in algorithm 4, in which we iteratively minimize $J_S(x^S; Z^S)$ and $J_T(x^T; Z^T)$ with respect to the source- and target-language clusterings, while simultaneously enforcing the cross-lingual cluster constraints. In this way we can generate cross-lingual clusterings using all the monolingual data while still making sure that the clusters are consistent between both languages. We refer to the clusters induced with this method as X-LINGUAL CLUSTERS.

In practice we found that each unconstrained monolingual run of the exchange algorithm (lines 1 and 3) tends to move the clustering away from one that obeys the word alignment constraints, which makes the procedure prone to diverge. Fixing the clustering of the words that are assigned clusters in the projection stages (lines 2 and 4) and only clustering the remaining words give more stable results. However, we found that iterating the procedure has little effect on model transfer performance and we therefore set $N = 1$ for all subsequent experiments. In this case the monolingual target language model is only used to expand the projected clustering, which has the effect of increasing coverage to word types in the target language that do not occur in the much smaller bitext.

8.4 Cross-Lingual Experiments

We next conduct an empirical evaluation of cross-lingual word-cluster features for cross-lingual model transfer. As in the first part of the chapter, we consider the use of these clusters for both syntactic dependency parsing and named-entity recognition.

8.4.1 Experimental Setup

In our first set of experiments on using cross-lingual cluster features, we evaluate the use of model transfer of a parser trained with English as the source language to the ten Indo-European languages listed in section 8.2. The English source-parser is trained on Stanford-style dependencies (De Marneffe et al., 2006). We exclude Korean and Chinese from this study, as initial experiments proved model transfer a poor technique when transferring parsers between such diverse languages. However, see chapter 9 for methods that facilitate transfer between more diverse languages.

We study the impact of using cross-lingual cluster features by comparing the strong delexicalized baseline model of McDonald et al. (2011), which only has features derived from universal part-of-speech tags, projected from English with the method of Petrov et al. (2012), to the same model when adding features derived from cross-lingual clusters. In both cases the feature models are the same as those used in section 8.2.2, except that they are delexicalized by removing all lexical word-identity features. We evaluate both the PROJECTED CLUSTERS and the X-LINGUAL CLUSTERS.

For these experiments we train the averaged structured perceptron for only five epochs. This is done in order to prevent over-fitting, which is an acute problem in this setting, due to the divergence between the source and target languages. Furthermore, in accordance with standard practice in studies on cross-lingual transfer, we only evaluate the unlabeled attachment score (UAS, see section 4.1.2). This is due to the fact that each treebank uses a different — possibly non-overlapping — label set, as discussed in chapter 7.

In our second set of experiments, we evaluate model transfer of a named-entity recognition system trained on English to German, Spanish and Dutch. We use the same feature models as in the monolingual case, with the exception that we use universal part-of-speech tags, automatically predicted with the method of Das and Petrov (2011), for all languages. Furthermore, the capitalization feature is removed when transferring from English to German. Capitalization is both a prevalent and a highly predictive feature of named entities in English, Spanish and Dutch. However, it is even more prevalent in German, but has very low predictive power, since all nouns in German are capitalized.

Interestingly, while delexicalization has shown to be important for model transfer of dependency parsers (McDonald et al., 2011), we noticed in preliminary experiments that it substantially degrades performance for named-entity recognition. We hypothesize that this is because word features are predictive of common proper names and that these are often translated directly across languages, at least in the case of newswire text. As for the transfer parser, when training the source named-entity recognition model, we regularize the model more heavily by setting the regularization hyper-parameter to $\lambda = 10$.

Table 8.5. *Results of model transfer for dependency parsing from English in terms of unlabeled attachment score (UAS).* ONLY SUBJECT/OBJECT *– UAS measured only over words marked as subject/object in the evaluation data.*

	ALL RELATIONS			ONLY SUBJECT/OBJECT		
Lang.	NO CL.	PROJECTED	X-LINGUAL	NO CL.	PROJECTED	X-LINGUAL
da	36.7	**38.9**	38.7	44.6	**49.8**	49.2
de	48.9	50.3	**50.7**	56.7	57.1	**59.0**
el	59.5	61.1	**63.0**	67.2	72.2	**72.5**
es	60.2	62.6	**62.9**	60.7	**65.9**	**65.9**
fr	70.0	71.6	**72.1**	77.4	80.4	**80.9**
it	64.6	68.6	**68.8**	64.6	70.5	**72.7**
nl	52.8	**54.5**	54.3	59.5	**67.0**	65.7
pt	66.8	70.7	**71.0**	53.3	**62.6**	62.5
ru	29.7	32.9	**34.4**	29.3	34.6	**37.2**
sv	55.4	**57.0**	56.9	57.3	**65.0**	64.4
avg	54.5	56.8	**57.3**	57.1	62.5	**63.0**

All word alignments for the cross-lingual clusterings were produced with the dual-decomposition aligner of DeNero and Macherey (2011) using 10.5 million (Danish) to 12.1 million (French) sentences of general aligned web data, from sources such as news text, consumer review sites, and so on.

8.4.2 Results

Table 8.5 lists the results of the transfer experiments for dependency parsing. The baseline results are comparable to those in McDonald et al. (2011) and thus also significantly outperform the results of recent unsupervised approaches (Berg-Kirkpatrick and Klein, 2010; Naseem et al., 2010). Importantly, cross-lingual cluster features are helpful across the board and give a relative error reduction ranging from 3% for Danish to 13% for Portuguese, with an average reduction of 6%, in terms of unlabeled attachment score (UAS). This shows the utility of cross-lingual cluster features for syntactic transfer. However, X-LINGUAL CLUSTERS provides roughly the same performance as PROJECTED CLUSTERS suggesting that even simple methods of cross-lingual clustering are sufficient for model transfer dependency parsing.

Remember from the discussion in chapter 7 that due to the divergences between source and target language annotation style, these results are likely to be under-estimating the parsers' actual ability to predict Stanford-style dependencies in the target languages. To highlight this point we run two additional experiments. First, linguists who are also fluent speakers of German, re-annotated the German test set to make the annotation consistent with Stanford-style dependencies. Using this data, NO CLUSTERS obtains 60.0%

Table 8.6. *Results of model transfer for named-entity recognition from English in terms of average F_1-score on the CoNLL 2002/2003 development and test sets.*

	de	es	nl	avg
NO CLUSTERS	25.4	49.5	49.9	41.6
PROJECTED CLUSTERS	39.1	62.1	61.8	54.4
X-LINGUAL CLUSTERS	**43.1**	**62.8**	**64.7**	**56.9**
↑ DEVELOPMENT SET ↓ TEST SET				
NO CLUSTERS	23.5	45.6	48.4	39.1
PROJECTED CLUSTERS	35.2	59.1	56.4	50.2
X-LINGUAL CLUSTERS	**40.4**	**59.3**	**58.4**	**52.7**

UAS, PROJECTED CLUSTERS 63.6% and X-LINGUAL CLUSTERS 64.4%. When compared to the scores on the original data set (48.9%, 50.3% and 50.7%, respectively) we can see not only that the baseline system is doing much better, but that the improvement from cross-lingual clustering is much more pronounced. Next, we investigated the accuracy of subject and object dependencies, as these are often annotated in similar ways across treebanks, typically modifying the main verb of the sentence. The right half of table 8.5 shows the scores when restricted to such dependencies in the gold data. We measure the percentage of dependents in subject and object relations that modify the correct word. Indeed, here we see the difference in performance become clearer, with the cross-lingual cluster model reducing errors by 14% relative to the non-cross-lingual model and upwards of 22% relative for Italian. These findings support the hypothesis that the results in table 8.5 are under-estimating the effectiveness of the transfer models.

We now turn to the results of the transfer experiments for named-entity recognition, shown in table 8.6. While the performance of the transfer systems is very poor when no word clusters are used, adding cross-lingual word clusters give substantial improvements across all languages. The simple PROJECTED CLUSTERS work well, but the X-LINGUAL CLUSTERS provide even larger improvements. On average the latter reduce errors on the test set by 22% in terms of F_1 and up to 26% for Spanish. We also measure how well the transferred named-entity recognition systems are able to detect entity boundaries (ignoring the entity categories). Here, on average, the best clusters provide a 24% relative error reduction on the test set (75.8 versus 68.1 F_1).

To our knowledge there are no comparable results on transfer learning of named-entity recognition systems. Although the results of the transfer systems are substantially below those of fully supervised systems, two points should be raised: First, the comparison to supervised results might be slightly unfavorable, since while the domains in the supervised case are in perfect alignment, this is not the case for the transfer systems (although all data sets studied here belong to the genre of news text). Second, looking at the results

of entity boundary detection of the transfer systems (based on evaluations not included here), it seems plausible that the label-specific predictions could be substantially improved by using manually, or automatically, constructed language specific gazetteers.

9. Target Language Adaptation of Discriminative Transfer Parsers

In the previous chapter, we studied the use of cross-lingual word clusters as a way to extend the delexicalized features typically employed in cross-lingual model transfer. While these features were shown to be useful for cross-lingual transfer of both syntactic dependency parsers and named-entity recognizers, the method used for inducing the cross-lingual clusters relies on the availability of bitext (or potentially other resources such as bilingual dictionaries). This somewhat restrict its applicability, as such resources may not always be available in large enough quantities, if at all. Furthermore, for a given target language, bitext is typically available only with a small number of source languages (predominantly English) and as we observed in the previous chapter, transfer across typologically divergent languages tend to not work very well.

In contrast, this chapter considers extensions to delexicalized model transfer which do not rely on any bitext or other bilingual resources. Specifically, we consider the multi-source transfer setting briefly discussed in section 7.2.3, focusing on methods for target language adaptation of delexicalized discriminative graph-based dependency parsers. We first show how recent insights on selective parameter sharing, based on typological and language-family features, can be applied to discriminative parsers by carefully decomposing the features of the parser. We then show how the parser can be relexicalized and adapted using unlabeled target language data and a learning method that can incorporate diverse knowledge sources through ambiguous labelings. In the latter scenario, we exploit two sources of knowledge: arc marginals derived from the base parser in a self-training algorithm, and arc predictions from multiple transfer parsers in an ensemble-training algorithm. Our final model outperforms the state of the art in multi-source transfer parsing on 15 out of 16 evaluated languages.

9.1 Multi-Source Delexicalized Transfer

Learning with multiple languages has been shown to benefit unsupervised learning (Cohen and Smith, 2009; Snyder and Barzilay, 2010; Berg-Kirkpatrick and Klein, 2010). Annotations in multiple languages can be combined in delexicalized transfer as well, as long as the parser features are available

Table 9.1. *Typological features from WALS (Dryer and Haspelmath, 2011) proposed for selective sharing by Naseem et al. (2012). Feature 89A has the same value for all studied languages, while 88A differs only for Basque. Both of these features are therefore subsequently excluded.*

Feature	Description
81A	Order of Subject, Object and Verb
85A	Order of Adposition and Noun
86A	Order of Genitive and Noun
87A	Order of Adjective and Noun
88A	Order of Demonstrative and Noun
89A	Order of Numeral and Noun

across the involved languages. This idea was explored by McDonald et al. (2011), who showed that target language accuracy can be improved by simply concatenating delexicalized treebanks in multiple languages. In similar work, Cohen et al. (2011) proposed a mixture model in which the parameters of a generative target language parser is expressed as a linear interpolation of source language parameters, whereas Søgaard (2011) showed that target side language models can be used to selectively subsample training sentences from the source languages to improve accuracy. Recently, inspired by the phylogenetic prior of Berg-Kirkpatrick and Klein (2010), Søgaard and Wulff (2012) proposed — among other ideas — a typologically informed weighting heuristic for linearly interpolating source language parameters. However, this weighting did not provide significant improvements over uniform weighting.

The aforementioned approaches work well for transfer between similar languages. However, their assumptions cease to hold for typologically divergent languages; a target language can rarely be described as a linear combination of data or model parameters from a set of source languages, as languages tend to share varied typological traits. This critical insight is discussed further in section 9.3.

To account for this issue, Naseem et al. (2012) recently introduced a novel generative model of dependency parsing, in which the generative process is factored into separate steps for the *selection* of dependents and their *ordering*. The parameters used in the selection step are all language independent, capturing only head-dependent attachment preferences. In the ordering step, however, parameters are *selectively shared* between subsets of source languages based on typological features of the languages extracted from WALS — the World Atlas of Language Structures (Dryer and Haspelmath, 2011).[1] Table 9.1 lists the typological features used, while table 9.2 shows the values of these features for the languages studied by Naseem et al., as well as in the

[1]WALS is available at http://wals.info/ — December 9, 2012.

Table 9.2. *Values of the typological features from WALS for the studied languages.*

Lang.	81A	85A	86A	87A
ar	VSO	Prepositions	Noun-Genitive	Noun-Adjective
bg	SVO	Prepositions	No dominant	Adjective-Noun
ca	SVO	Prepositions	Noun-Genitive	Noun-Adjective
cs	SVO	Prepositions	No dominant	Adjective-Noun
de	No dominant	Prepositions	Noun-Genitive	Adjective-Noun
el	No dominant	Prepositions	Noun-Genitive	Adjective-Noun
en	SVO	Prepositions	No dominant	Adjective-Noun
es	SVO	Prepositions	Noun-Genitive	Noun-Adjective
eu	SOV	Postpositions	Genitive-Noun	Noun-Adjective
hu	No dominant	Postpositions	Genitive-Noun	Adjective-Noun
it	SVO	Prepositions	Noun-Genitive	Noun-Adjective
ja	SOV	Postpositions	Genitive-Noun	Adjective-Noun
nl	No dominant	Prepositions	Noun-Genitive	Adjective-Noun
pt	SVO	Prepositions	Noun-Genitive	Noun-Adjective
sv	SVO	Prepositions	Genitive-Noun	Adjective-Noun
tr	SOV	Postpositions	Genitive-Noun	Adjective-Noun
zh	SVO	No dominant	Genitive-Noun	Adjective-Noun

experiments presented later in this chapter. In the transfer scenario, where no supervision is available in the target language, this parser achieves state-of-the-art-performance across a number of languages; in particular for target languages with a word order divergent from the source languages.

However, the generative model of Naseem et al. is quite impoverished. In the fully supervised setting, it obtains substantially lower accuracies compared to a standard arc-factored graph-based parser (McDonald et al., 2005). On average, over 16 languages,[2] the generative model trained with full supervision on the target language obtains an accuracy of 67.1%. A comparable lexicalized discriminative arc-factored model (McDonald et al., 2005) obtains 84.1%. Even when delexicalized, this model achieves 78.9%. As shown in table 9.3, this gap in supervised accuracy holds for all 16 languages. Thus, while selective sharing is a powerful device for transferring parsers across languages, the underlying generative model used by Naseem et al. (2012) restricts its potential performance.

9.2 Basic Models and Experimental Setup

Inspired by the superiority of discriminative graph-based parsing in the supervised scenario, we investigate whether the insights of Naseem et al. (2012) on selective parameter sharing can be incorporated into such models in the

[2]Based on the results presented in Naseem et al. (2012), excluding English.

Table 9.3. *Supervised accuracies of the generative selective sharing model (Table 2, column "MLE" in Naseem et al. (2012)) versus discriminative arc-factored log-linear models with delexicalized features and lexicalized features.*

		Arc-Factored	
Language	Generative	Delexicalized	Lexicalized
ar	64.2	74.0	82.6
bg	71.0	86.5	90.0
ca	72.1	86.3	91.7
cs	58.9	68.8	83.4
de	58.0	83.0	86.6
el	70.5	79.3	83.1
es	65.3	78.7	84.3
eu	51.6	64.8	68.7
hu	61.6	77.1	79.9
it	72.3	81.7	83.9
ja	75.6	84.2	92.2
nl	58.0	70.8	76.3
pt	79.6	86.2	88.3
sv	73.0	84.0	87.3
tr	67.6	73.2	78.4
zh	73.5	83.6	89.3
avg	67.1	78.9	84.1

transfer scenario. This section reviews the basic graph-based parser framework and the experimental setup that will be used throughout. Section 9.3 delves into details on how to incorporate selective sharing in this model. Finally, section 9.4 describes how learning with ambiguous labelings in this parser can be used to further adapt the parser to the target language, both through self-training and through ensemble-training.

9.2.1 Discriminative Graph-Based Parser

Recall the discriminative arc-factored parsing model from section 3.2.2 and the discriminative probabilistic formulation described in section 3.4. Let $x \in \mathcal{X}$ denote an input sentence and let $y \in \mathcal{Y}(x)$ denote a dependency tree, where $\mathcal{Y}(x)$ is the set of well-formed trees spanning x. In what follows, we will restrict $\mathcal{Y}(x)$ to be the set of projective trees spanning x, but all our methods are equally applicable in the non-projective case.[3] Provided a vector of model parameters θ, the probability of a dependency tree $y \in \mathcal{Y}(x)$, conditioned on

[3]We do however hypothesize that projectivity constraints may be useful for guiding learning in the case of ambiguous supervision, as discussed in section 9.4.1.

a sentence x, has the standard log-linear conditional form:

$$p_\theta(y \mid x) = \frac{\exp\left\{\theta^\top \Phi(x, y)\right\}}{\sum_{y' \in \mathcal{Y}(x)} \exp\left\{\theta^\top \Phi(x, y')\right\}}\,.$$

Without loss of generality, we restrict ourselves to first-order models, where the feature function $\Phi(x, y)$ factors over individual arcs in y, such that

$$\Phi(x, y) = \sum_{(h,d) \in y} \phi(x, h, d)\,,$$

where $h \in [0, |x|]$ and $d \in [1, |x|]$ are the indices of the head word and the dependent word of the arc, respectively, with $h = 0$ representing a dummy ROOT token. Taking log loss as the surrogate loss function for empirical risk minimization (see section 4.1.3), the model parameters are estimated by minimizing the negative log-likelihood of the training data $\mathcal{D} = \{(x^{(j)}, y^{(j)})\}_{j=1}^{m}$:

$$J(\theta; \mathcal{D}) = -\sum_{j=1}^{m} \log p_\theta(y^{(j)} \mid x^{(j)})\,.$$

The standard gradient-based L-BFGS algorithm (Liu and Nocedal, 1989) is used to maximize the log-likelihood, while Eisner's algorithm (Eisner, 1996) is used for inference of the Viterbi parse and arc-marginals. When using delexicalized features there is no need to apply a regularizer (see section 4.1.1) as these features are too coarse for the model to overfit.

9.2.2 Data Sets and Experimental Setup

To facilitate comparison with the state of the art, we use the same treebanks and experimental setup as Naseem et al. (2012). Notably, we use the mapping proposed by Naseem et al. (2010) to map from fine-grained treebank specific part-of-speech tags to coarse-grained "universal" tags, rather than the more recent mapping proposed by Petrov et al. (2012). For each target language evaluated, the treebanks of the remaining languages are used as *labeled* training data, while the target language treebank is used for testing only (in section 9.4 a different portion of the target language treebank is additionally used as *unlabeled* training data). We refer the reader to Naseem et al. (2012) for detailed information on the different treebanks. Due to divergent treebank annotation guidelines, which makes fine-grained evaluation difficult, all results are evaluated in terms of unlabeled attachment score (UAS). Following Naseem et al. (2012), we use gold part-of-speech tags and evaluate only on sentences of length 50 or less excluding punctuation. The two-letter abbreviations from the ISO 639-1 standard (see appendix A) are used when referring to the languages in tables.

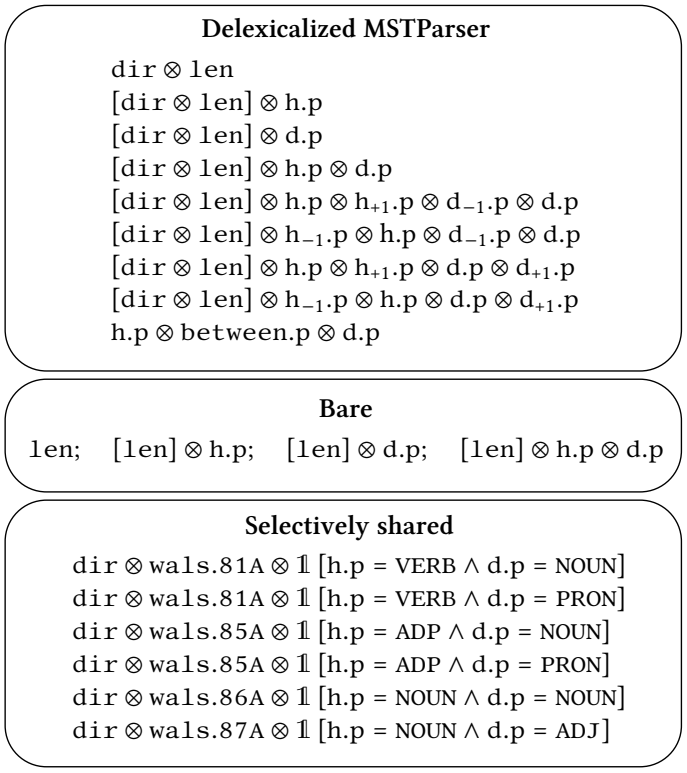

Figure 9.1. Feature templates used in the arc-factored parser models. Direction: dir ∈ {LEFT, RIGHT}. Dependency length: len ∈ {1, 2, 3, 4, 5+}. Part of speech of head / dependent / words between head and dependent: h.p / d.p / between.p ∈ {NOUN, VERB, ADJ, ADV, PRON, DET, ADP, NUM, CONJ, PRT, PUNC, X}; token to the left / right of x: x_{-1} / x_{+1}; WALS features: wals.Y for Y = 81A, 85A, 86A, 87A (see tables 9.1 and 9.2). [·] denotes an optional template; for example, [dir ⊗ len] ⊗ h.p ⊗ d.p expands to templates dir ⊗ len ⊗ h.p ⊗ d.p and h.p ⊗ d.p, that is, the template also falls back on its undirectional variant.

9.2.3 Baseline Models

We compare our models to two multi-source baseline models. The first baseline, *NB&G*, is the generative model with selective parameter sharing from Naseem et al. (2012).[4] This model is trained without target language data, but we investigate the use of such data in section 9.4.2. The second baseline, *Delex*, is a delexicalized projective version of the well-known graph-based MSTParser (McDonald et al., 2005). The feature templates used by this model are shown in the top of fig. 9.1. Note that there is no selective sharing in this model.

[4] Model "D-,T_o" in Table 2 from Naseem et al. (2012).

The second and third columns of table 9.4 show the unlabeled attachment scores of the baseline models for each target language. We see that *Delex* performs well on target languages that are related to the majority of the source languages. However, for languages that diverge from the Indo-European majority family, the selective sharing model, *NB&G*, achieves substantially higher accuracies.

9.3 Feature-Based Selective Sharing

The results for the baseline models are not surprising considering the feature templates used by *Delex*. There are two fundamental issues with these features when used for direct transfer. First, all but one template include the arc direction. Second, several features are sensitive to local word order. As an example, consider the feature template

$$[\texttt{dir} \otimes \texttt{len}] \otimes \mathrm{h.p} \otimes \mathrm{h}_{+1}\mathrm{.p} \otimes \mathrm{d}_{-1}\mathrm{.p} \otimes \mathrm{d.p}\,,$$

which models direction (`dir`) and length (`len`), as well as word order in the local contexts of the head ($\mathrm{h.p} \otimes \mathrm{h}_{+1}\mathrm{.p}$) and the dependent ($\mathrm{d}_{-1}\mathrm{.p} \otimes \mathrm{d.p}$). Such features do not transfer well across typologically diverse languages.

In order to verify that these issues are the cause of the poor performance of the *Delex* model, we remove all directional features and all features that model local word order from *Delex*. The feature templates of the resulting *Bare* model are shown in the middle of fig. 9.1. These features only model selectional preferences and dependency length, analogously to the selection component of Naseem et al. (2012). The performance of *Bare* is shown in the fourth column of table 9.4. The removal of most of the features results in a performance drop on average. However, for languages outside of the Indo-European family, *Bare* is often more accurate, especially for Basque, Hungarian and Japanese, which supports our hypothesis.

9.3.1 Sharing Based on Typological Features

After removing all directional features, we now carefully reintroduce them. Inspired by Naseem et al. (2012), we make use of the typological features from WALS (Dryer and Haspelmath, 2011), listed in table 9.1, to selectively share directional parameters between languages. As a natural first attempt at sharing parameters, one might consider forming the cross-product of all features of *Delex* with all WALS properties, similarly to a common domain adaptation technique (Daumé III, 2007; Finkel and Manning, 2009). However, this approach has two issues. First, it results in a huge number of features, making the model prone to overfitting. Second, and more critically, it ties together languages via features for which they are not typologically similar.

Table 9.4. *Unlabeled attachment scores of the multi-source transfer models. Boldface numbers indicate the best result per language. Underlined numbers indicate languages whose group is not represented in the training data (these default to Share under Similarity and Family). NB&G is the "D-,T_o" model in Table 2 from Naseem et al. (2012).*

		Discriminative Arc-Factored Models				
Language	NB&G	Delex	Bare	Share	Similar	Family
ar	**57.2**	43.3	43.1	52.7	52.7	52.7
bg	**67.6**	64.5	56.1	65.4	62.4	65.4
ca	71.9	72.0	58.1	66.1	**80.2**	77.6
cs	43.9	40.5	43.1	42.5	**45.3**	43.5
de	54.0	57.0	49.3	55.2	58.1	**59.2**
el	61.9	63.2	57.7	62.9	59.9	**63.2**
es	62.3	66.9	52.6	59.3	**69.0**	67.1
eu	39.7	29.5	43.3	**46.8**	**46.8**	**46.8**
hu	56.9	56.2	60.5	**64.5**	**64.5**	**64.5**
it	68.0	70.8	55.7	63.5	**74.6**	72.5
ja	62.3	38.9	50.6	57.1	64.6	**65.9**
nl	56.2	**57.9**	51.6	55.0	51.8	56.8
pt	76.2	77.5	63.0	72.7	**78.4**	**78.4**
sv	52.0	61.4	55.9	58.8	48.8	**63.5**
tr	59.1	37.4	36.0	41.7	**59.5**	59.4
zh	**59.9**	45.1	47.9	54.8	54.8	54.8
avg	59.3	55.1	51.5	57.4	60.7	**62.0**

Consider English and French, which are both prepositional and thus have the same value for WALS property 85A. Among other features, these languages would end up sharing a parameter for the feature

$$[\texttt{dir} \otimes \texttt{len}] \otimes \mathbb{1}\,[\texttt{h.p} = \textsc{noun}] \otimes \mathbb{1}\,[\texttt{d.p} = \textsc{adj}] \otimes \texttt{wals.85A};$$

yet they have the exact opposite direction of attachment preference when it comes to nouns and adjectives. This problem applies to any parameter mixing method that treats all the parameters as equal.

Like Naseem et al. (2012), we instead share parameters more selectively. Our strategy is to use the relevant part-of-speech tags of the head and dependent to select which parameters to share, based on very basic linguistic knowledge. The resulting features are shown in the bottom of fig. 9.1. For example, there is a shared directional feature that models the order of Subject, Object and Verb by conjoining WALS feature 81A with the arc direction and an indicator feature that fires only if the head is a verb and the dependent is a noun. These features would not be very useful by themselves, so we combine them with the *Bare* features. The accuracy of the resulting *Share* model is shown in column five of table 9.4. Although this model still performs

worse than *NB&G*, it is an improvement over the *Delex* baseline and actually outperforms the former on 5 out of the 16 languages.

9.3.2 Sharing Based on Language Groups

While *Share* models selectional preferences and arc directions for a subset of dependency relations, it does not capture the rich local word order information captured by *Delex*. We now consider two ways of selectively including such information based on language similarity. While more complex sharing could be explored (Berg-Kirkpatrick and Klein, 2010), we use a flat structure and consider two simple groupings of the source and target languages.

First, the *Similar* model consists of the features used by *Share* together with the features from *Delex* in fig. 9.1. The latter are conjoined with an indicator feature that fires only when the source and target languages share values for all the WALS features in table 9.1. This is accomplished by adding the template

$$\texttt{f} \otimes [\texttt{wals.81A} \otimes \texttt{wals.85A} \otimes \texttt{wals.86A} \otimes \texttt{wals.87A} \otimes \texttt{wals.88A}]$$

for each feature template `f` in *Delex*. This groups: 1) Catalan, Italian, Portuguese and Spanish; 2) Bulgarian, Czech and English; 3) Dutch, German and Greek; and 4) Japanese and Turkish. The remaining languages do not share all WALS properties with at least one source language and thus revert to *Share*, since they cannot exploit these grouped features.

Second, instead of grouping languages according to WALS, the *Family* model is based on a simple subdivision into Indo-European languages (Bulgarian, Catalan, Czech, Greek, English, Spanish, Italian, Dutch, Portuguese, Swedish) and Altaic languages (Japanese, Turkish). This is accomplished with indicator features analogous to those used in *Similar*. The remaining languages are again treated as isolates and revert to *Similar*.

The results for these models are given in the last two columns of table 9.4. We see that by adding these rich features back into the fold, but having them fire only for languages in the same group, we can significantly increase the performance — from 57.4% to 62.0% on average when considering *Family*. If we consider our original *Delex* baseline, we see an absolute improvement of 6.9% on average and a relative error reduction of 15%. Particular gains are seen for non-Indo-European languages. For example, Japanese increases from 38.9% to 65.9%. Furthermore, *Family* achieves a 7% relative error reduction over the *NB&G* baseline and outperforms it on 12 of the 16 languages. This shows that a discriminative graph-based parser can achieve higher accuracies compared to generative models when the features are carefully constructed.

9.4 Target Language Adaptation

While some higher-level linguistic properties of the target language have been incorporated through selective sharing, so far no features specific to the target language have been employed. Cohen et al. (2011) and Naseem et al. (2012) have shown that using expectation-maximization (EM) to this end can in some cases bring substantial accuracy gains. For discriminative models, self-training has been shown to be quite effective for adapting monolingual parsers to new domains (McClosky et al., 2006), as well as for *relexicalizing* delexicalized parsers using unlabeled target language data (Zeman and Resnik, 2008). Similarly Täckström (2012) used self-training to adapt a multi-source cross-lingual named-entity recognizer to different target languages, "relexicalizing" the model with word cluster features native to the target language. However, as discussed in the next section, standard self-training is suboptimal for such target language adaptation.

9.4.1 Ambiguity-Aware Training

In this section, we propose a related training method based on ambiguous supervision; see section 5.1. In this setting a discriminative probabilistic model is induced from automatically inferred ambiguous labelings over unlabeled target language data, in place of gold-standard dependency trees. The ambiguous labelings can combine multiple sources of evidence to guide the estimation or simply encode the underlying uncertainty from the base parser. This uncertainty is marginalized out during training. The structure of the output space, for example, projectivity and single-headedness constraints, along with regularities in the feature space, can together guide the estimation, similar to what occurs with the expectation-maximization algorithm.

Core to this method is the idea of an *ambiguous labeling* $\tilde{\boldsymbol{y}}(x) \subset \mathcal{Y}(x)$, which encodes a set of possible dependency trees for an input sentence x. In subsequent sections we describe how to define such labelings. Critically, $\tilde{\boldsymbol{y}}(x)$ should be large enough to capture the correct labeling, but on the other hand small enough to provide concrete guidance for model estimation. Ideally, $\tilde{\boldsymbol{y}}(x)$ will capture heterogenous knowledge that can aid the parser in target language adaptation. In a first-order arc-factored model, we define $\tilde{\boldsymbol{y}}(x)$ in terms of a collection of *ambiguous arc sets* $\mathcal{A}(x) = \{\mathcal{A}(x, d)\}_{d=1}^{|x|}$, where $\mathcal{A}(x, d)$ denotes the set of ambiguously specified heads for the dth token in x. Then, $\tilde{\boldsymbol{y}}(x)$ is defined as the set of all projective dependency trees spanning x that can be assembled from the arcs in $\mathcal{A}(x)$.

Methods for learning with ambiguous labelings have previously been proposed in the context of multi-class classification (Jin and Ghahramani, 2002), sequence-labeling (Dredze et al., 2009), log-linear lexical-functional grammar (LFG) parsing (Riezler et al., 2002), as well as for discriminative reranking of generative constituency parsers (Charniak and Johnson, 2005). In contrast to

Dredze et al. (2009), who allow weights to be assigned to partial labels, we assume that the ambiguous arcs are weighted uniformly. For target language adaptation, these weights would typically be derived from unreliable sources and we do not want to train the model to simply mimic their beliefs. Furthermore, with this assumption we can directly apply the latent log loss (see section 5.2.1), which means that learning is achieved by minimizing the negative *marginal* log-likelihood of the ambiguous training set $\widetilde{\mathcal{D}} = \{(x^{(j)}, \tilde{\boldsymbol{y}}(x^{(j)})\}_{j=1}^{m}$:

$$J(\theta; \widetilde{\mathcal{D}}) \;= -\sum_{j=1}^{m} \log \left\{ \sum_{y \in \tilde{\boldsymbol{y}}(x^{(j)})} p_\theta(y \mid x^{(j)}) \right\} + \lambda \, \|\theta\|_2^2 \,.$$

In minimizing the negative marginal log-likelihood, the model is free to distribute probability mass among the trees in the ambiguous labeling to its liking, as long as the marginal log-likelihood improves. The same objective function was used by Riezler et al. (2002) and Charniak and Johnson (2005). A key difference is that in these works, the ambiguity is constrained through a supervised signal, while ambiguity is here used as a way to achieve self-training, leveraging the base-parser itself, or some other potentially noisy knowledge source as the sole constraints. Note that we have introduced an ℓ_2-regularizer, weighted by λ. This is important as we are now training *lexicalized* target language models which can easily overfit. In all experiments, we optimize parameters with L-BFGS; see section 4.2.1. Recall that the negative marginal log-likelihood is non-convex, so that we are only guaranteed to find a local minimum.

Ambiguity-aware self-training

In standard self-training — hereafter referred to as *Viterbi self-training* — a base parser is used to label each unlabeled sentence with its most probable parse tree to create a self-labeled data set, which is subsequently used to train a supervised parser. There are two reasons why this simple approach may work. First, if the base parser's errors are not too systematic and if the self-training model is not too expressive, self-training can reduce the variance on the new domain. Second, self-training allows for features in the new domain with low support — or no support in the case of lexicalized features — in the base parser to be "filled in" by exploiting correlations in the feature representation. However, a potential pitfall of this approach is that the self-trained parser is encouraged to blindly mimic the base parser, which leads to error reinforcement. This may be particularly problematic when relexicalizing a transfer parser, since the lexical features provide the parser with increased power and thereby an increased risk of overfitting to the noise. To overcome this potential problem, we propose an *ambiguity-aware self-training* (AAST) method that is able to take the noise of the base parser into account.

We use the arc-marginals of the base parser to construct the ambiguous labeling $\tilde{\boldsymbol{y}}(x)$ for a sentence x. For each token $d \in [1, |x|]$, we first sort the set

of arcs where d is the dependent, $\{(h, d)\}_{h=0}^{|x|}$, by the marginal probabilities of the arcs:

$$p_\theta(h, d \mid x) = \sum_{\{\, y : y \in \mathcal{Y}(x) \wedge (h,d) \in y \,\}} p_\theta(y \mid x)\,.$$

We next construct the ambiguous arc set $\mathcal{A}(x, d)$ by adding arcs (h, d) in order of decreasing probability, until their cumulative probability exceeds σ, that is, until

$$\sum_{(h,d) \in \mathcal{A}(x,d)} p_\theta(h, d \mid x) \geq \sigma\,.$$

Lower values of σ result in more aggressive pruning, with $\sigma = 0$ corresponding to including no arc and $\sigma = 1$ corresponding to including all arcs. We always add the highest scoring tree $\hat{y}$ to $\tilde{\boldsymbol{y}}(x)$ to ensure that it contains at least one complete projective tree.

Figure 9.2 outlines an example of how (and why) AAST works. In the Greek example, the genitive phrase Η παραμονή σκαφών (*the stay of vessels*) is incorrectly analyzed as a flat noun phrase. This is not surprising given that the base parser simply observes this phrase as DET NOUN NOUN. However, looking at the arc marginals we can see that the correct analysis is available during AAST, although the actual marginal probabilities are quite misleading. Furthermore, the genitive noun σκαφών also appears in other less ambiguous contexts, where the base parser correctly predicts it to modify a noun and not a verb. This allows the training process to add weight to the corresponding lexical feature pairing σκαφών with a noun head and away from the feature pairing it with a verb. The resulting parser correctly predicts the genitive construction.

Ambiguity-aware ensemble-training

While ambiguous labelings can be used as a means to improve self-training, any information that can be expressed as hard arc-factored constraints can be incorporated, including linguistic expert knowledge and annotation projected via bitext. Here we explore another natural source of information: the predictions of other transfer parsers. It is well known that combining several diverse predictions in an ensemble often leads to improved predictions. However, in most ensemble methods there is typically no learning involved once the base learners have been trained (Sagae and Lavie, 2006). An exception is the method of Sagae and Tsujii (2007), who combine the outputs of many parsers on unlabeled data to train a parser for a new domain. However, in that work the learner is not exposed to the underlying ambiguity of the base parsers; it is only given the Viterbi parse of the combination system as the gold standard. In contrast, we propose an *ambiguity-aware ensemble-training* (AAET) method that treats the union of the ensemble predictions for a sentence x as an ambiguous labeling $\tilde{\boldsymbol{y}}(x)$. An additional advantage of this

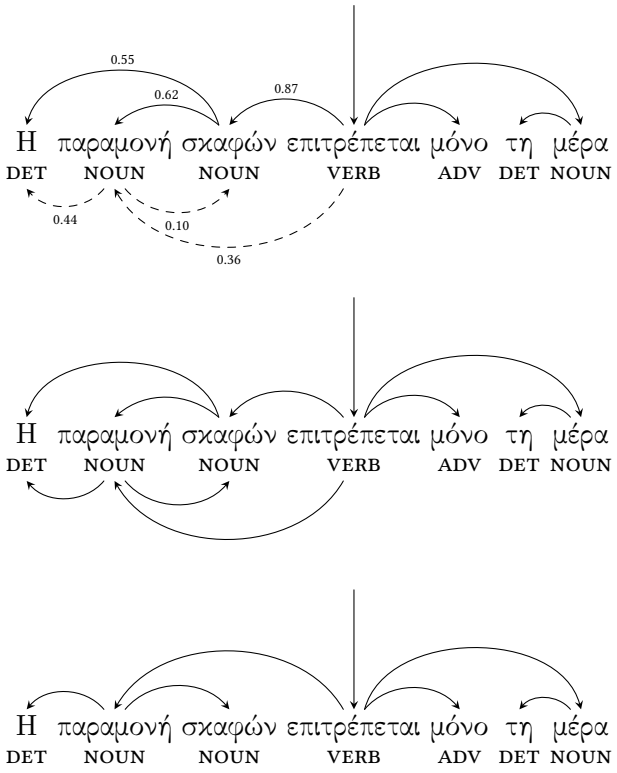

Figure 9.2. An example of ambiguity-aware self-training (AAST) on a sentence from the Greek self-training data. The sentence roughly translates to *The stay of vessels is permitted only for the day.* **Top**: Arcs from the base model's Viterbi parse ($\hat{y}(x)$) are shown above the sentence. When only the part-of-speech tags are observed, the parser tends to treat everything to the left of the verb as a head-final noun phrase. The dashed arcs below the sentence are the correct arcs for the true genitive construction *stay of vessels*. These arcs and the corresponding incorrect arcs in the Viterbi parse are marked with their marginal probabilities. **Middle**: The ambiguous labeling $\tilde{\boldsymbol{y}}(x)$, which is used as supervision in AAST. Additional non-Viterbi arcs are present in $\tilde{\boldsymbol{y}}(x)$; for clarity, these are not shown. When learning with AAST, probability mass will be pushed towards any tree consistent with $\tilde{\boldsymbol{y}}(x)$. Marginal probabilities are ignored at this stage, so that all arcs in $\tilde{\boldsymbol{y}}(x)$ are treated as equals. **Bottom**: The Viterbi parse of the AAST model, which has selected the correct arcs from $\tilde{\boldsymbol{y}}(x)$.

approach is that the ensemble is compiled into a single model and therefore does not require multiple models to be stored and used at runtime.

It is straightforward to construct $\tilde{\boldsymbol{y}}(x)$ from multiple parsers. First, let $\mathcal{A}_k(x, d)$ be the set of arcs for the dth token in x according to the kth parser in the ensemble. When arc-marginals are used to construct the ambiguity set for the kth parser, $|\mathcal{A}_k(x, d)| \geq 1$, whereas when the Viterbi-parse is used, $\mathcal{A}_k(x, d)$ is a singleton. Next form $\mathcal{A}(x, d) = \bigcup_k \mathcal{A}_k(x, d)$ as the ensemble arc ambiguity set from which $\tilde{\boldsymbol{y}}(x)$ is assembled. In this study, the arc sets of two base parsers are combined: first, the arc-marginal ambiguity set of the base parser and second, the Viterbi arc set from the *NB&G* parser of Naseem et al. (2012) in table 9.4.[5] Thus, the latter will have singleton arc ambiguity sets, but when combined with the arc-marginal ambiguity sets of our base parser, the result will encode uncertainty derived from both parsers.

9.4.2 Adaptation Experiments

We now study the different approaches to target language adaptation empirically. As in Naseem et al. (2012), we use the CoNLL training sets, stripped of all dependency information, as the unlabeled target language data in our experiments. We use the *Family* model as the base parser, which is used to label the unlabeled target data with the Viterbi parses as well as with the ambiguous labelings. The final model is then trained on this data using standard lexicalized features (McDonald et al., 2005). Since labeled training data is unavailable in the target language, we cannot tune any hyper-parameters and simply set $\lambda = 1$ and $\sigma = 0.95$ throughout.[6] Although the latter may suggest that $\tilde{\boldsymbol{y}}(x)$ contains a high degree of ambiguity, in reality, the marginal distributions of the base model have low entropy and after filtering with $\sigma = 0.95$, the average number of potential heads per dependent ranges from 1.4 to 3.2, depending on the target language.

The ambiguity-aware training methods — ambiguity-aware self-training *(AAST)* and ambiguity-aware ensemble-training *(AAET)* — are compared to three baseline systems. First, *NB&G+EM* is the generative model of Naseem et al. (2012) trained with expectation-maximization on additional unlabeled target language data. Second, *Family* is the best model from the previous section. Third, *Viterbi* is the basic Viterbi self-training model. The results of each of these models across languages are shown in table 9.5.

There are a number of things that can be observed. First, *Viterbi* self-training helps slightly on average, but the gains are not consistent and there are even drops in accuracy for some languages. Second, *AAST* outperforms

[5] We do not have access to the marginals of *NB&G*; otherwise they could be included as well.

[6] Since we assume the availability of multiple source languages, it would in principle be possible to perform cross-validation on held-out *source* language data. However, for simplicity, we simply fix these parameters a priori and use the same setting for all target languages.

Table 9.5. *Results of the target-language adaptation experiments. AAST: ambiguity-aware self-training. AAET: ambiguity-aware ensemble-training. Boldface numbers indicate the best result per language. Underlined numbers indicate the best result, excluding AAET. NB&G+EM is the "D+,T_o" model from Naseem et al. (2012).*

			Target Language Adaptation		
Language	NB&G+EM	Family	Viterbi	AAST	AAET
ar	**59.3**	52.7	52.6	53.5	58.7
bg	67.0	65.4	66.4	67.9	**73.0**
ca	71.7	77.6	78.0	**79.9**	76.1
cs	44.3	43.5	43.6	44.4	**48.3**
de	54.1	59.2	59.7	**62.5**	61.5
el	67.9	63.2	64.5	65.5	**69.6**
es	62.0	67.1	68.2	**68.5**	66.9
eu	47.8	46.8	47.5	48.6	**49.4**
hu	58.6	64.5	64.6	65.6	**67.5**
it	65.6	72.5	71.6	72.4	**73.4**
ja	64.1	65.9	65.7	68.8	**72.0**
nl	56.6	56.8	57.9	58.1	**60.2**
pt	75.8	78.4	79.9	**80.7**	79.9
sv	61.7	63.5	63.4	**65.5**	**65.5**
tr	59.4	59.4	59.5	64.1	**64.2**
zh	51.0	54.8	54.8	57.9	**60.7**
avg	60.4	62.0	62.4	64.0	**65.4**

the *Viterbi* variant on all languages and nearly always improves on the base parser, although it sees a slight drop for Italian. *AAST* improves the accuracy over the base model by 2% absolute on average and by as much as 5% absolute for Turkish. Comparing this model to the *NB&G+EM* baseline, we observe an improvement by 3.6% absolute, outperforming it on 14 of the 16 languages. Furthermore, ambiguity-aware self-training appears to help more than expectation-maximization for generative (unlexicalized) models. Naseem et al. observed an increase from 59.3% to 60.4% on average by adding unlabeled target language data and the gains were not consistent across languages. *AAST*, on the other hand, achieves consistent gains, rising from 62.0% to 64.0% on average. Third, as shown in the rightmost column of table 9.5, ambiguity-aware ensemble-training is indeed a successful strategy — *AAET* outperforms the previous best self-trained model on 13 and *NB&G+EM* on 15 out of 16 languages. The relative error reduction with respect to the base *Family* model is 9% on average, while the average reduction with respect to *NB&G+EM* is 13%.

Before concluding, two additional points are worth making. First, further gains may potentially be achievable with feature-rich discriminative models. While the best generative transfer model of Naseem et al. (2012) approaches

its upper-bounding supervised accuracy (60.4% vs. 67.1%), our relaxed self-training model is still far below its supervised counterpart (64.0% vs. 84.1%). One promising statistic along these lines is that the oracle accuracy for the ambiguous labelings of *AAST* is 75.7%, averaged across languages, which suggests that other training algorithms, priors or constraints could improve the accuracy substantially. Second, relexicalization is a key component of self-training. If we use delexicalized features during self-training, we only observe a small average improvement from 62.0% to 62.1%.

10. Token and Type Constraints for Part-of-Speech Tagging

The previous two chapters focused on cross-lingual model transfer; via cross-lingual word clusters in chapter 8 and via selective parameter sharing combined with ambiguity-aware training in chapter 9. Part of the motivation for considering model transfer in these chapters was the application of these methods to linguistic structure spanning multiple tokens. As discussed in chapter 7, this is a setting where annotation projection is less likely to succeed, due to both cross-lingual divergences and to alignment errors. For structure defined at the token level, such as parts of speech, annotation projection may on the other hand be more viable. However, even for such structure, typically only parts of the target-side structure will be specified by the projected annotation and these parts are still susceptible to transfer noise.

Based on these considerations, this chapter turns to cross-lingual learning of part-of-speech taggers. To conquer the incomplete and noisy nature of the *token*-level constraints specified by the projected annotation, we propose to incorporate additional *type*-level constraints derived from crowdsourced lexica. While prior work has successfully considered both token- and type-level projection across word-aligned bitext for estimating the model parameters of generative tagging models (Yarowsky and Ngai, 2001; Xi and Hwa, 2005), a key observation underlying the approach described in this chapter is that token- and type-level information offer different and complementary signals. On the one hand, token-level annotation projected via high-confidence alignments offer precise constraints on a tag in a particular context, while type-level constraints merely specify the parts of speech that a word may possibly embody in some context. On the other hand, manually created type-level dictionaries can have broad coverage and do not suffer from word-alignment errors. They can therefore be used to filter both systematic and random token-level projection noise; as we shall see, such filtering is necessary for achieving good performance with projected token-level tags.

By coupling these constraints, an ambiguous labeling of the target language text is achieved, which can subsequently be used as supervision for a discriminative conditional random field (CRF) model with latent variables; see section 5.2.1. As discusses in chapter 3, the use of a discriminative model makes it possible to incorporate arbitrary features over the input. In addition to standard (contextual) lexical features and tag transition features, monolingual target language word-cluster features (see chapter 8) are shown to be

particularly useful in this setting. Most of these features can also be used for training a generative log-linear hidden Markov model (HMM, see section 5.2.1) with ambiguous supervision. However, the best results are obtained with the discriminative CRF model. This is an important finding, as most previous work on weakly supervised part-of-speech tagging has been restricted to generative models.

To evaluate the proposed approach, empirical results are presented and analyzed for standard publicly available data sets in 15 languages: eight Indo-European languages previously studied in this context by Das and Petrov (2011) and Li et al. (2012), together with 7 additional languages from different families, for which no comparable study has been performed.

10.1 Token and Type Constraints

Type-level information has been amply used in ambiguously supervised part-of-speech tagging, either via manually crafted tag dictionaries (Smith and Eisner, 2005a; Ravi and Knight, 2009; Garrette and Baldridge, 2012, 2013), noisily projected tag dictionaries (Das and Petrov, 2011), or through crowdsourced lexica, such as Wiktionary (Li et al., 2012).[1] At the other end of the spectrum, there have been attempts at projecting token-level information across word-aligned bitext (Yarowsky and Ngai, 2001; Xi and Hwa, 2005). However, systems that combine both sources of information in a single model have yet to be fully explored. The following three subsections provides a brief outline of token and type constraints and our overall approach for coupling these two types of information to build robust part-of-speech taggers that do not require any direct supervision in the target language.

10.1.1 Token Constraints

Fully supervised approaches to part-of-speech tagging are by definition based on complete token-level constraints. Although a recent study by Garrette and Baldridge (2013) suggests that annotating word types rather than tokens can in some instances be more useful, at least when the time allotted for annotation is severely limited, token-level constraints are the most powerful source of supervision available as it is the only level which permits full disambiguation of linguistic structure in context. Therefore, we should strive towards obtaining token-level information, as long as it can be obtained at sufficiently high quality.

For many resource-poor languages, which lack native token-level supervision, there is at least some bitext with a resource-rich source language available.[2] It is then natural to consider using an annotation projection approach,

[1] http://www.wiktionary.org/ — April 6, 2013.

[2] For simplicity, we choose English as our source language in all subsequent experiments.

where a supervised tagger is used to predict the parts of speech of the tokens in the source side of the bitext, which are subsequently projected to the target side via automatic word alignments; see chapter 7 for an overview of cross-lingual annotation projection methods. As discussed in section 7.2.1, various approaches to handle the noise resulting from this process have been proposed. Importantly, attempts that use only one source of projected annotation or that use the same source of information to filter out the projection noise are only able to filter out random noise, as the systematic bias that arise from different source language annotations and syntactic divergences still remains.

10.1.2 Type Constraints

It is well known that given a tag dictionary, even if it is incomplete, it is possible to learn accurate part-of-speech taggers (Smith and Eisner, 2005a; Goldberg et al., 2008; Ravi and Knight, 2009; Naseem et al., 2009). While widely differing in their respective model structures and learning objectives, all of these approaches achieve excellent results, although they still perform substantially below fully supervised approaches. Unfortunately, they all rely on tag dictionaries extracted directly from the underlying treebank data. Such dictionaries provide in-depth coverage of the test domain and they moreover list all inflected word forms; such knowledge is difficult to obtain and unrealistic to expect for resource-poor languages.[3]

In contrast, Das and Petrov (2011) automatically create type-level tag dictionaries by aggregating over projected token-level information extracted from bitext. To handle the noise in these automatic dictionaries, they use label propagation on a similarity graph to smooth (and also expand) the label distributions. While their approach produces good results and is applicable to resource-poor languages, it requires a complex multi-stage training procedure including the construction of a large distributional similarity graph.

Recently, Li et al. (2012) presented a simple and viable alternative: crowdsourced dictionaries from Wiktionary. While noisy and sparse in nature, Wiktionary dictionaries are available for 170 languages.[4] Furthermore, their quality and coverage is growing continuously (Li et al., 2012). By incorporating type constraints from Wiktionary into a log-linear HMM (Berg-Kirkpatrick et al., 2010), Li et al. (2012) were able to obtain the best published results in

[3] A number of approaches to learning part-of-speech taggers with neither token-level supervision nor tag dictionaries has been considered (Brill et al., 1990; Brill and Marcus, 1992; Brown et al., 1992; Finch and Chater, 1992; Schütze, 1993, 1995; Maron et al., 2010; Lamar et al., 2010; Dhillon et al., 2012); see also the survey by Christodoulopoulos et al. (2010) and the recent PASCAL shared task on unsupervised grammar induction (Gelling et al., 2012). These approaches all perform substantially below approaches that utilize tag-dictionaries.

[4] http://meta.wikimedia.org/wiki/Wiktionary — April 6, 2013.

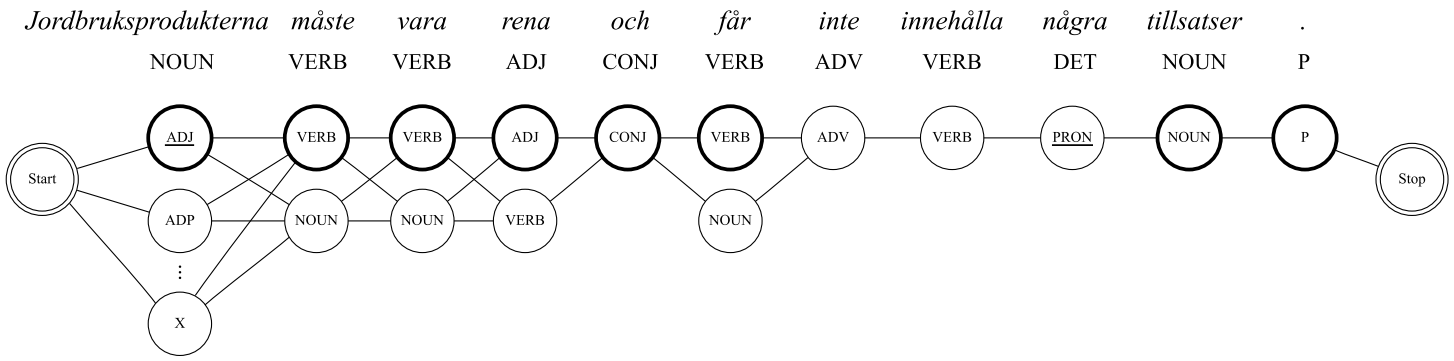

Figure 10.1. Lattice representation of the inference search space $\mathcal{Y}(x)$ for an authentic sentence in Swedish (*The farming products must be pure and must not contain any additives*), after pruning with type constraints derived from Wiktionary. The correct parts of speech are listed underneath each word. Bold nodes indicate projected token constraints $\tilde{y}$. The coupled constraints lattice $\widehat{\mathcal{Y}}(x, \tilde{y})$ correspond to the bold nodes together with nodes for words that are lacking token constraints. In this case, the coupled constraints lattice thus defines exactly one valid path. Underlined text indicates incorrect part-of-speech tags in the constraints lattice.

this setting, surpassing the results of Das and Petrov (2011) on eight Indo-European languages.

10.1.3 Coupled Token and Type Constraints

Rather than relying exclusively on either token or type constraints, we propose to leverage the complementary strengths of both sources of information during training. For each sentence in our training set, a partially constrained lattice of tag sequences is constructed as follows:

1. For each token whose type is *not* in the tag dictionary, we allow the entire tag set.
2. For each token whose type *is* in the tag dictionary, we prune all tags not licensed by the dictionary and mark the token as dictionary-pruned.
3. For each token that has a tag projected via a high-confidence bidirectional word alignment: if the projected tag is still present in the lattice, then we prune every tag but the projected tag for that token; if the projected tag is not present in the lattice, which can only happen for dictionary-pruned tokens, then we ignore the projected tag.

Note that only step 3 requires word-aligned bitext. Figure 10.1 provides a running example. The lattice shows tags permitted after constraining the words to tags licensed by the dictionary (up until Step 2 above). There is only a single token *Jordbruksprodukterna* (*the farming products*) not in the dictionary; in this case the lattice permits the full set of tags. With token-level projections (Step 3; nodes with bold border in fig. 10.1), the lattice can be further pruned. In most cases, the projected tag is both correct and is in the dictionary-pruned lattice. We thus successfully disambiguate such tokens and shrink the search space substantially.

There are two cases we highlight in order to show where our model can break. First, for the token *Jordbruksprodukterna*, the erroneously projected tag ADJ will eliminate all other tags from the lattice, including the correct tag NOUN. Second, the token *några* (*any*) has a single dictionary entry PRON and is missing the correct tag DET. In the case where DET is the projected tag, we will not add it to the lattice and simply ignore it. This is because we hypothesize that the tag dictionary can be trusted more than the tags projected via noisy word alignments. As we will see in section 10.3, taking the union of tags performs worse, which supports this hypothesis.

For the discriminative CRF model, we need to define two lattices: one that the model moves probability mass towards and another one defining the overall search space (or partition function); see section 10.2.2. In traditional supervised learning without a dictionary, the former is a trivial lattice containing the gold standard tag sequence and the latter is the set of all possible tag sequences spanning the tokens. With our best model, we will move mass towards the coupled token- and type-constrained lattice, such that the model

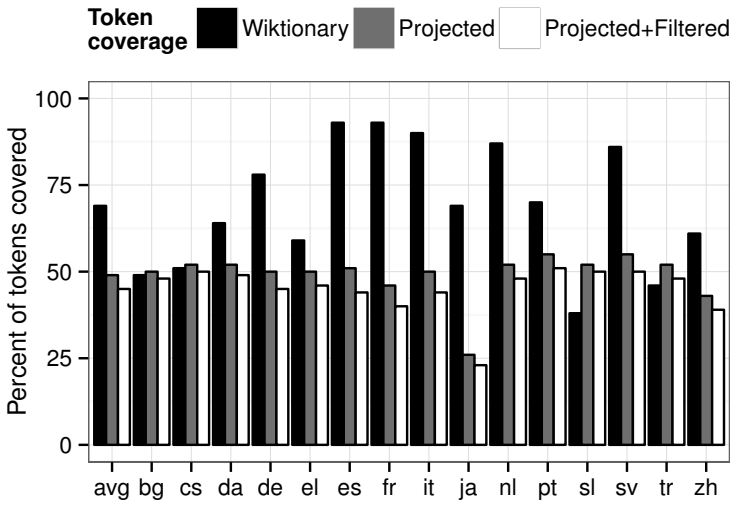

Figure 10.2. Wiktionary and projection dictionary coverage per language. The bars show the percentage of tokens in the target side of the bitext that are covered by Wiktionary (black), that have a projected tag (grey), and that have a projected tag after intersecting the two (white).

can freely distribute mass across all paths consistent with these constraints. The lattice defining the partition function consists of the full set of possible tag sequences when no dictionary is used. When a dictionary is used it consists of all dictionary-pruned tag sequences (that is, the lattice achieved by skipping Step 3 above). The full set of possibilities are shown in fig. 10.1 for our running example.

For the generative HMM model, we need to define only one lattice; see section 10.2.1. For our best generative model, this turns out to be the coupled token- and type-constrained lattice. At prediction time, in both the discriminative and the generative cases, we find the most likely label sequence using Viterbi inference; see section 3.5.

As further motivation, let us look at what token- and type-level coverage we can achieve with our proposed approach. Figures 10.2 and 10.3 provide statistics regarding the supervision coverage and remaining ambiguity when coupling type constraints from Wiktionary with projected token constraints. Figure 10.2 shows that more than two thirds of all tokens in our training data belong to word types that are covered by Wiktionary; see section 10.3.1 for details on the data and the version of Wiktionary used. However, there is considerable variation between languages: Spanish has the highest coverage with over 90%, while Turkish, an agglutinative language with a vast number of word forms, has less than 50% coverage. Figure 10.3 shows that there is substantial uncertainty left after pruning with Wiktionary, since tokens are rarely fully disambiguated: 1.3 tags per token are allowed on average for

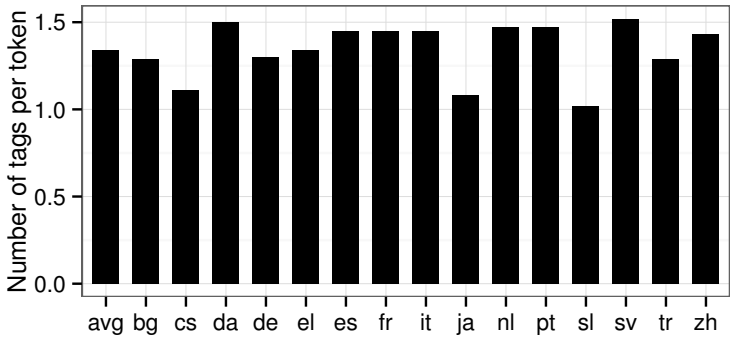

Figure 10.3. Average number of Wiktionary-licensed tags per token on the target side of the bitext, for types in Wiktionary.

word types in Wiktionary. Of course, for tokens of types not in Wiktionary, the tag ambiguity is substantially higher — for the coarse tag set considered in this chapter, there are 12 potential tags per token.

Figure 10.2 further shows that high-confidence alignments are available for about half of the tokens for most languages (Japanese is a notable exception with less than 30% of the tokens covered). Intersecting the Wiktionary tags and the projected tags (Step 2 and 3 above) filters out a small fraction of potentially erroneous tags, but preserves the majority of the projected tags; the remaining, more accurate projected tags cover almost half of all tokens, greatly reducing the search space that the learner needs to explore.

10.2 Models with Coupled Constraints

Let us describe more formally how the token and type constraints are coupled and how the resulting coupled constraints are used to train probabilistic tagging models. Recalling the notation from chapter 3, let $x = (x_1 x_2 \dots x_{|x|}) \in \mathcal{X}$ denote a sentence, where each token $x_i \in \mathcal{V}$ is an instance of a word type from the vocabulary $\mathcal{V}$ and let $y = (y_1 y_2 \dots y_{|x|}) \in \mathcal{Y}$ denote a tag sequence, where $y_i \in \mathcal{T}$ is the tag assigned to token x_i and $\mathcal{T}$ denotes the set of possible part-of-speech tags. We denote the lattice of all admissible tag sequences for the sentence x by $\mathcal{Y}(x)$. This is the inference search space in which the tagger operates. As our empirical results will show, it is crucial to constrain the size of this lattice in order to guide the model during training. Here, we achieve this by means of coupled token and type constraints.

Type constraints are provided by a tag dictionary, which maps a word type $x_i \in \mathcal{V}$ to a set of admissible tags $\mathcal{T}(x_i) \subseteq \mathcal{T}$. For word types not in the dictionary we allow the full set of tags $\mathcal{T}$ (while possible, in this chapter we do not attempt to distinguish closed-class versus open-class words). When provided with a tag dictionary, the lattice of admissible tag sequences for a

sentence x is given by $\mathcal{Y}(x) = \mathcal{T}(x_1) \times \mathcal{T}(x_2) \times \ldots \times \mathcal{T}(x_{|x|})$. In the scenario where no tag dictionary is available, the inference space is instead given by the full lattice $\mathcal{Y}(x) = \mathcal{T}^{|x|}$.

Type constraints can drastically reduce the inference space. However, as shown in fig. 10.3, a significant amount of ambiguity remains even after pruning with the dictionary. To disambiguate further, we therefore consider projected token-level constraints. Let $\tilde{y} = (\tilde{y}_1 \tilde{y}_2 \ldots \tilde{y}_{|x|})$ be the projected tags for the sentence x, such that $\{\tilde{y}_i\} = \emptyset$ for tokens without a projected tag. Note that we here assume that each token has at most one projected tag. This is not a fundamental restriction and we hypothesize that it may be possible to achieve better results by maintaining some ambiguity in the projection step. Indeed, recall from chapter 9 that at least in the context of graph-based syntactic dependency parsing, preserving ambiguity for training with noisy inputs can be beneficial.

Next, we define a piecewise operator $\frown$ that couples $\tilde{y}$ and $\mathcal{Y}(x)$ with respect to every sentence index, resulting in a token- and type-constrained lattice. The operator behaves as follows, coherent with the high-level description in section 10.1.3:

$$\widehat{\mathcal{T}}(x_i, \tilde{y}_i) \;= \tilde{y}_i \frown \mathcal{T}(x_i) = \begin{cases} \{\tilde{y}_i\} & \text{if } \{\tilde{y}_i\} \neq \emptyset \wedge \tilde{y}_i \in \mathcal{T}(x_i) \\ \mathcal{T}(x_i) & \text{otherwise} \end{cases}.$$

We denote the token- and type-constrained lattice by $\widehat{\mathcal{Y}}(x, \tilde{y}) = \widehat{\mathcal{T}}(x_1, \tilde{y}_1) \times \widehat{\mathcal{T}}(x_2, \tilde{y}_2) \times \ldots \times \widehat{\mathcal{T}}(x_{|x|}, \tilde{y}_{|x|})$. Note that when token-level projections are not used, the dictionary-pruned lattice and the lattice with coupled constraints are identical, that is, $\widehat{\mathcal{Y}}(x, \tilde{y}) = \mathcal{Y}(x)$.

The next subsections describe the structured probabilistic latent variable models that we consider for training with type, token, and coupled type and token constraints. We have already seen the definition of these models in section 3.4, whereas training of these models with ambiguous labelings was described in section 5.2.2. Below, we recapitulate these models in terms of the above constrained lattices.

10.2.1 HMMs with Coupled Constraints

Recall the HMM model with log-linear emission and transition distributions from section 3.4.2. A first-order HMM,[5] defines the joint distribution of a sentence $x \in \mathcal{X}$ and a tag-sequence $y \in \mathcal{Y}(x)$ as

$$p_\beta(x, y) \;= \prod_{i=1}^{|x|} \underbrace{p_\beta(x_i \mid y_i)}_{\text{emission}} \underbrace{p_\beta(y_i \mid y_{i-1})}_{\text{transition}},$$

[5] Although Li et al. (2012) observed improved results with a second-order HMM in the type-supervised setting, for simplicity, we will subsequently only consider first-order models.

where the emission and transition factors are both local log-linear models.

The lattice $\widehat{\mathcal{Y}}(x, \tilde{y})$ corresponds to an ambiguous labeling. As described in chapter 5, the latent log loss is therefore a good fit for training. Equivalently, we seek to maximize the likelihood of the observed parts of the data, marginalizing over the ambiguity. For this we need the joint marginal distribution $p_\beta(x, \widehat{\mathcal{Y}}(x, \tilde{y}))$ of a sentence x, and its coupled constraints lattice $\widehat{\mathcal{Y}}(x, \tilde{y})$, which has the simple form

$$p_\beta(x, \widehat{\mathcal{Y}}(x, \tilde{y})) \;=\; \sum_{y \in \widehat{\mathcal{Y}}(x, \tilde{y})} p_\beta(x, y)\,.$$

If there are no projections and no tag dictionary, then $\widehat{\mathcal{Y}}(x, \tilde{y}) = \mathcal{T}^{|x|}$, and consequently $p_\beta(x, \widehat{\mathcal{Y}}(x, \tilde{y})) = p_\beta(x)$, which reduces to fully unsupervised learning. Given a constrained training set $\mathcal{D} = \{(x^{(j)}, \tilde{y}^{(j)})\}_{j=1}^m$, where $\{\tilde{y}_i^{(j)}\} = \emptyset$ for tokens $x_i^{(j)}$ whose tag is unconstrained, our objective is to minimize the ℓ_2-regularized negative *marginal joint* log-likelihood

$$J(\beta; \mathcal{D}) \;=\; -\sum_{j=1}^{m} \log p_\beta(x^{(j)}, \widehat{\mathcal{Y}}(x^{(j)}, \tilde{y}^{(j)})) + \lambda \left\|\beta\right\|_2^2 . \tag{10.1}$$

We take a direct gradient approach for optimizing eq. (10.1), as described in section 5.2.2.[6] Recall that since the negative marginal log-likelihood is non-convex, we are only guaranteed to find a local minimum of the objective function.

10.2.2 CRFs with Coupled Constraints

Recall the CRF model from section 3.4.1. Again, restricting ourselves to a first-order model, the conditional probability of a tag sequence $y \in \mathcal{Y}(x)$ for a sentence $x \in \mathcal{X}$ is given by

$$p_\theta(y \mid x) \;=\; \frac{\exp\left\{\theta^\top \Phi(x, y)\right\}}{\sum_{y' \in \mathcal{Y}(x)} \exp\left\{\theta^\top \Phi(x, y')\right\}}\,,$$

where $\mathcal{Y}(x)$ is the dictionary-pruned lattice *without* the token constraints when a tag dictionary is used. Again, as described in section 5.2.2 we seek to minimize the latent log loss of the conditional model, or equivalently to maximize the marginal conditional log-likelihood of the observed parts of the data. For this, we need to marginalize over all sequences consistent with the lattice $\widehat{\mathcal{Y}}(x, \tilde{y})$:

$$p_\theta(\widehat{\mathcal{Y}}(x, \tilde{y}) \mid x) \;=\; \sum_{y \in \widehat{\mathcal{Y}}(x, \tilde{y})} p_\theta(y \mid x)\,.$$

[6] We trained the HMM with EM as well, but achieved better results with the direct gradient approach. We therefore only use the direct gradient approach for our empirical study.

Note that if there are no projections and no tag dictionary, $p_\theta(\widehat{\mathcal{Y}}(x, \tilde{y}) \mid x) = 1$ and no learning is possible in this model. In our empirical study, we will see that it is crucial to keep the level of ambiguity of the observations low when training with this conditional model. Given the constrained training set $\mathcal{D}$, the parameters of the constrained CRF are estimated by minimizing the ℓ_2-regularized negative *marginal conditional* log-likelihood of the constrained data:

$$J(\theta; \mathcal{D}) \;= -\sum_{j=1}^{m} \log p_\theta(\widehat{\mathcal{Y}}(x^{(j)}, \tilde{y}^{(j)}) \mid x^{(j)}) + \lambda \|\theta\|_2^2 \,. \tag{10.2}$$

Again, since the negative marginal conditional log-likelihood is non-convex, we are only guaranteed to find a local minimum of eq. (10.2).

10.3 Empirical Study

We now present a detailed empirical study of the models proposed in the previous sections. In addition to comparing with the state of the art in Das and Petrov (2011) and Li et al. (2012), we present models with several combinations of token and type constraints, additional features incorporating word clusters, and explore both generative and discriminative models.

10.3.1 Experimental Setup

Before delving into the experimental details and results, a description is provided of the experimental setup in terms of corpora, tag set, bitext, Wiktionary definitions, model features and optimization settings.

Languages

We evaluate on the eight target languages used in the previous work of Das and Petrov (2011) and Li et al. (2012): Danish (da), German (de), Greek (el), Spanish (es), Italian (it), Dutch (nl), Portuguese (pt) and Swedish (sv). In addition, we add the following seven languages: Bulgarian (bg), Czech (cs), French (fr), Japanese (ja), Slovene (sl), Turkish (tr) and Chinese (zh). Since the former eight languages all belong to the Indo-European family, we thereby broaden the coverage to language families more distant from the source language. For all languages, English is used as the source language. We use the treebanks from the CoNLL shared tasks on dependency parsing (Buchholz and Marsi, 2006; Nivre et al., 2007) for evaluation.[7] The two-letter abbreviations from the ISO 639-1 standard (in parentheses above; see appendix A) are used when referring to these languages in tables and figures.

[7] For French we use the treebank of Abeillé et al. (2003).

Tag set
In all cases, the language-specific part-of-speech tags are mapped to universal tags using the mapping of Petrov et al. (2012).[8] Since we use ambiguous supervision via projected tags or Wiktionary, and since the number of model states equals the number of tags, the model states induced by all models correspond one-to-one to part-of-speech tags. This allows us to compute tagging accuracy without a greedy one-to-one or many-to-one mapping.

Bitext
For all experiments, we use English as the source language. Depending on availability, there are between 1M and 5M parallel sentences for each language. The majority of the parallel data is gathered automatically from the web using the method of Uszkoreit et al. (2010). We further include data from Europarl (Koehn, 2005) and from the UN parallel corpus (UN, 2006), for languages covered by these corpora. The English side of the bitext is tagged with parts of speech, using a standard supervised CRF tagger, trained on the Penn Treebank (Marcus et al., 1993), with tags mapped to universal tags. The parallel sentences are word aligned with the aligner of DeNero and Macherey (2011). Intersected high-confidence alignments (confidence > 0.95) are extracted and aggregated into projected type-level dictionaries. For purely practical reasons, the training data with token-level projections is created by randomly sampling target-side sentences with a total of 500K tokens.

Wiktionary
We use a snapshot of the Wiktionary word definitions, and follow the heuristics of Li et al. (2012) for creating the Wiktionary dictionary, by mapping the Wiktionary tags to universal part-of-speech tags.[9]

Features
For all models, we use only an identity feature for tag-pair transitions. We use five features that couple the current tag and the observed word (analogous to the emission in an HMM): word identity, suffixes of up to length 3, and three indicator features that fire when the word starts with a capital letter, contains a hyphen or contains a digit. These are the same features as those used by Das and Petrov (2011). Finally, for some models we add a word cluster feature that couples the current tag and the word cluster identity of the word. The same (monolingual) clustering model as in chapter 8 is used in these experiments; again setting the number of clusters to 256 across all languages. The clusters for each language are learned on a large monolingual newswire corpus.

[8] We use version 1.03 of the mappings available at http://code.google.com/p/universal-pos-tags — February 12, 2013.

[9] The definitions were downloaded from http://toolserver.org/~enwikt/definitions — August 31, 2012. This snapshot is more recent than that used by Li et al. (2012).

Table 10.1. *Tagging accuracies for type-constrained HMM models.* $\mathcal{Y}^{HMM}$ *is the "Feature-HMM" model (trained without any type-constraints) and D&P is the "With LP" model, both from Table 2 of Das and Petrov (2011), while LG&T is the "SHMM-ME" model in Table 2 of Li et al. (2012).* $\mathcal{Y}^{HMM}_{proj.}$, $\mathcal{Y}^{HMM}_{wik.}$ *and* $\mathcal{Y}^{HMM}_{union}$ *are HMMs trained solely with type constraints derived from the projected dictionary, from Wiktionary and from the union of these dictionaries, respectively.* $\mathcal{Y}^{HMM}_{union}$*+C is equivalent to* $\mathcal{Y}^{HMM}_{union}$ *with additional word cluster features. All models are trained on the treebank of each language, stripped of gold labels. Results are averaged over the eight languages common to the studies of Das and Petrov (2011) and Li et al. (2012), denoted* avg (8)*, as well as over the full set of 15 languages, denoted* avg.

	Prior work			HMM with type constraints			
Lang.	$\mathcal{Y}^{\text{HMM}}$	D&P	LG&T	$\mathcal{Y}^{\text{HMM}}_{\text{proj.}}$	$\mathcal{Y}^{\text{HMM}}_{\text{wik.}}$	$\mathcal{Y}^{\text{HMM}}_{\text{union}}$	$\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C
bg	–	–	–	84.2	68.1	87.2	**87.9**
cs	–	–	–	75.4	70.2	75.4	**79.2**
da	69.1	83.2	83.3	87.7	82.0	78.4	**89.5**
de	81.3	82.8	85.8	86.6	85.1	80.0	**88.3**
el	71.8	82.5	79.2	83.3	83.8	**86.0**	83.2
es	80.2	84.2	86.4	83.9	83.7	**88.3**	87.3
fr	–	–	–	**88.4**	75.7	75.6	86.6
it	68.1	86.8	86.5	89.0	85.4	89.9	**90.6**
ja	–	–	–	45.2	**76.9**	74.4	73.7
nl	65.1	79.5	**86.3**	81.7	79.1	83.8	82.7
pt	78.4	87.9	84.5	86.7	79.0	83.8	**90.4**
sl	–	–	–	78.7	64.8	82.8	**83.4**
sv	70.1	80.5	86.1	80.6	85.9	85.9	**86.7**
tr	–	–	–	**66.2**	44.1	65.1	65.7
zh	–	–	–	59.2	**73.9**	63.2	73.0
avg (8)	73.0	83.4	84.8	84.9	83.0	84.5	**87.3**
avg	–	–	–	78.5	75.9	80.0	**83.2**

Optimization settings

For all experiments, we use the L-BFGS batch algorithm (see section 4.2.1) to optimize the objective functions in eqs. (10.1) and (10.2). Since there is a large number of models to train, we only run L-BFGS for 100 iterations for each model, rather than to convergence. Preliminary experiments showed that there is little to no improvement in accuracy after this number of iterations. Since we lack labeled data to tune the regularization parameter, we simply set $\lambda = 1$ for all models.

10.3.2 Type-Constrained Models

To examine the effect of type constraints in isolation, we experiment with the HMM, drawing constraints from three different dictionaries. Table 10.1

compares the performance of our models to the best results of Das and Petrov (2011) and Li et al. (2012), denoted D&P and LG&T, respectively. In addition to these type-constrained baselines, the fully unsupervised HMM model of Das and Petrov (2011), denoted $\mathcal{Y}^{\text{HMM}}$, is included for comparison. As in previous work, training is done exclusively on the training portion of each treebank, stripped of any manual linguistic annotation.

We first use all of our parallel data to generate projected tag dictionaries: the English part-of-speech tags are projected across word alignments and aggregated to tag distributions for each word type. As in Das and Petrov (2011), the distributions are then filtered with a threshold of 0.2 to remove noisy tags and to create an unweighted tag dictionary. We refer to this model as $\mathcal{Y}^{\text{HMM}}_{\text{proj.}}$. Its average accuracy of 84.9% on the eight languages is higher than the 83.4% of D&P and on par with LG&T (84.8%).[10] Our next model ($\mathcal{Y}^{\text{HMM}}_{\text{wik.}}$) simply draws type constraints from Wiktionary. It slightly underperforms LG&T (83.0%), presumably because they used a second-order HMM. As a simple extension to these two models, we take the union of the projected dictionary and Wiktionary to constrain an HMM, which we name $\mathcal{Y}^{\text{HMM}}_{\text{union}}$. This model performs somewhat worse on the eight Indo-European languages (84.5%), compared to $\mathcal{Y}^{\text{HMM}}_{\text{proj.}}$ (84.9%), but gives an improvement over the projected dictionary when evaluated across all 15 languages (80.0% versus 78.5%).

We next add monolingual cluster features to the model with the union dictionary. This model, $\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C, significantly outperforms all other type-constrained models, demonstrating the utility of word-cluster features.[11] For further exploration, we train the same model on the data sets containing 500K tokens sampled from the target side of the parallel data ($\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C+L); this is done to explore the effects of large data during training. From table 10.2, we find that training on these data sets result in an average accuracy of 87.2% which is comparable to the 87.3% reported for $\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C in table 10.1. This shows that the different source domain and amount of training data do not influence the performance of the HMM significantly. Compared to the basic fully unsupervised $\mathcal{Y}^{\text{HMM}}$, which is trained without any type constraints, $\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C achieves a relative error reduction of 53%, averaged across the eight Indo-European languages.

Finally, we train multiple CRF models where we treat type constraints as a partially observed lattice and use the full unpruned lattice for computing the partition function (results not shown in table 10.1). We observe similar trends in these results, but on average, accuracies are much lower compared to the type-constrained HMM models; the CRF model with the union dictionary along with cluster features achieves an average accuracy of 79.3% when

[10] This model corresponds to the weaker "No LP" model of Das and Petrov (2011), which eschews the label propagation step used by these authors after the projection and aggregation step. We found that label propagation was only beneficial when small amounts of bitext were available.

[11] Note that these are monolingual clusters. Bilingual clusters as described in chapter 8 might bring additional benefits.

Table 10.2. *Tagging accuracies for token-constrained and coupled token- and type-constrained models. +C: cluster features. +L: large (500k tokens) training sets with (partial) token-level projections. The best type-constrained model ($\mathcal{Y}^{HMM}_{union}$+C+L) is included for comparison. The remaining columns show HMM/CRF models with token constraints ($\tilde{y}\ldots$) and coupled token and type constraints ($\widehat{\mathcal{Y}}\ldots$). The latter use the projected dictionary ($\cdot_{proj.}$), Wiktionary ($\cdot_{wik.}$) and the union of these dictionaries ($\cdot_{union}$). The search spaces of $\widehat{\mathcal{Y}}\ldots$ are each pruned with the respective tag dictionary used to derive the coupled constraints. The difference between $\widehat{\mathcal{Y}}^{CRF}_{wik.}$+C+L and $\mathcal{Y}^{HMM}_{union}$+C+L is statistically significant ($^{**}p < 0.01$, $^{*}p < 0.015$) according to a paired bootstrap test (Efron and Tibshirani, 1993). Significance was not assessed for* avg *or* avg (8).

		Token constraints		HMM with coupled constraints			CRF with coupled constraints		
Lang.	$\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C+L	$\tilde{y}^{\text{HMM}}$+C+L	$\tilde{y}^{\text{CRF}}$+C+L	$\widehat{\mathcal{Y}}^{\text{HMM}}_{\text{proj.}}$+C+L	$\widehat{\mathcal{Y}}^{\text{HMM}}_{\text{wik.}}$+C+L	$\widehat{\mathcal{Y}}^{\text{HMM}}_{\text{union}}$+C+L	$\widehat{\mathcal{Y}}^{\text{CRF}}_{\text{proj.}}$+C+L	$\widehat{\mathcal{Y}}^{\text{CRF}}_{\text{wik.}}$+C+L	$\widehat{\mathcal{Y}}^{\text{CRF}}_{\text{union}}$+C+L
bg	87.7	77.9	84.1	84.5	83.9	86.7	86.0	**87.8**	85.4
cs	78.3	65.4	74.9	74.8	**81.1**	76.9	74.7	80.3**	75.0
da	87.3	80.9	85.1	87.2	85.6	88.1	85.5	**88.2***	86.0
de	87.7	81.4	83.3	85.0	89.3	86.7	84.4	**90.5****	85.5
el	85.9	81.1	77.8	80.1	87.0	83.9	79.6	**89.5****	79.7
es	**89.1****	84.1	85.5	83.7	85.9	88.0	85.7	87.1	86.0
fr	**88.4****	83.5	84.7	85.9	86.4	87.4	84.9	87.2	85.6
it	89.6	85.2	88.5	88.7	87.6	**89.8**	88.3	89.3	89.4
ja	72.8	47.6	54.2	43.2	76.1	70.5	44.9	**81.0****	68.0
nl	83.1	78.4	82.4	82.3	84.2	83.2	83.1	**85.9****	83.2
pt	89.1	84.7	87.0	86.6	88.7	88.0	87.9	**91.0****	88.3
sl	**82.4**	69.8	78.2	78.5	81.8	80.1	79.7	82.3	80.0
sv	86.1	80.1	84.2	82.3	87.9	86.9	84.4	**88.9****	85.5
tr	62.4	58.1	64.5	64.6	61.8	64.8	65.0	64.1**	**65.2**
zh	72.6	52.7	39.5	56.0	74.1	73.3	59.7	**74.4****	73.4
avg (8)	87.2	82.0	84.2	84.5	87.0	86.8	84.9	**88.8**	85.4
avg	82.8	74.1	76.9	77.6	82.8	82.3	78.2	**84.5**	81.1

trained on same data. This result is not surprising. First, the CRF's search space is fully unconstrained. Second, the dictionary only provides a weak set of observation constraints, which do not provide sufficient information to successfully train a discriminative model. However, as we will observe next, coupling the dictionary constraints with token-level information solves this problem.

10.3.3 Token-Constrained Models

We now proceed to add token-level information, focusing in particular on coupled token and type constraints. Since it is not possible to generate projected token constraints for our monolingual treebanks, all models in this subsection are trained on the 500K-tokens bitext data sets used at the end of the previous section. As a baseline, we first train HMM and CRF models that use *only* projected token constraints ($\tilde{y}^{\text{HMM}}$+C+L and $\tilde{y}^{\text{CRF}}$+C+L). As shown in table 10.2, these models underperform the best type-level model ($\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C+L).[12] This confirms that projected token constraints are not reliable on their own, which is in line with similar projection models previously examined by Das and Petrov (2011).

We then study models with coupled token and type constraints. These models use the same three dictionaries as used in section 10.3.2, but additionally couple the derived type constraints with projected token constraints; see the caption of table 10.2 for a list of these models. Note that since we only allow projected tags that are licensed by the dictionary (Step 3 of the lattice construction; see section 10.1.3), the actual token constraints used in these models vary with the different dictionaries.

From table 10.2, we see that coupled constraints are superior to token constraints, when used both with the HMM and the CRF. However, for the HMM, coupled constraints do not provide any benefit over type constraints alone, in particular when the projected dictionary or the union dictionary is used to derive the coupled constraints ($\widehat{\mathcal{Y}}^{\text{HMM}}_{\text{proj.}}$+C+L and $\widehat{\mathcal{Y}}^{\text{HMM}}_{\text{union}}$+C+L). We hypothesize that this is because these dictionaries (in particular the former) have the same bias as the token-level tag projections, so that the dictionary is unable to correct the systematic errors in the projections (see section 10.1.1). Since the token constraints are stronger than the type constraints in the coupled models, this bias may have a substantial impact. With the Wiktionary dictionary, the difference between the type-constrained and the coupled-constrained HMM is negligible: $\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C+L and $\widehat{\mathcal{Y}}^{\text{HMM}}_{\text{wik.}}$+C+L both average at an accuracy of 82.8%.

The CRF model, on the other hand, is able to take advantage of the complementary information in the coupled constraints, provided that the dictionary

[12] Note that to make the comparison fair vis-a-vis potential divergences in training domains, we compare to the best type-constrained model trained on the same 500K tokens training sets.

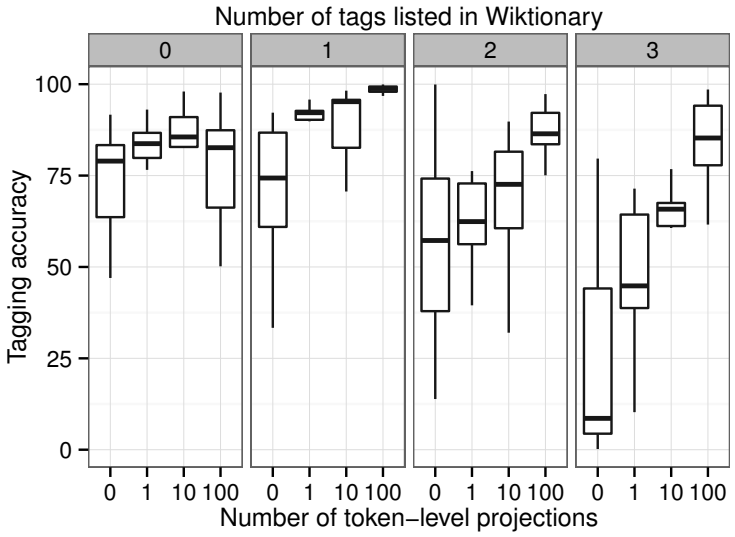

Figure 10.4. Relative influence of token and type constraints on tagging accuracy in the $\widehat{y}^{\text{CRF}}_{\text{wik.}}$+C+L model. Word types are categorized according to a) their number of Wiktionary tags (0,1,2 or 3+ tags, with 0 representing no Wiktionary entry; top-axis) and b) the number of times they are token-constrained in the training set (divided into buckets of 0, 1-9, 10-99 and 100+ occurrences; x-axis). The boxes summarize the accuracy distributions across languages for each word type category as defined by a) and b). The horizontal line in each box marks the median accuracy, the top and bottom mark the first and third quantile, respectively, while the whiskers mark the minimum and maximum values of the accuracy distribution.

is able to filter out the systematic token-level errors. With a dictionary derived from Wiktionary and projected token-level constraints, $\widehat{y}^{\text{CRF}}_{\text{wik.}}$+C+L performs better than all the remaining models, with an average accuracy of 88.8% across the eight Indo-European languages available to D&P and LG&T. Averaged over all 15 languages, its accuracy is 84.5%. Compared to the fully unsupervised y^{HMM}, the relative error reduction is 59%, averaged over the eight Indo-European languages.

10.3.4 Analysis

In this section, we provide a detailed analysis of the impact of token versus type constraints on $\widehat{y}^{\text{CRF}}_{\text{wik.}}$+C+L and study the pruning and filtering mistakes resulting from incomplete Wiktionary tag dictionary entries in detail. In order to not "contaminate" the treebank test sets with such detailed error analysis, this analysis is based on the training portion of each treebank. Note that the annotation of the training portion is only used for the purpose of analysis — as above, no target treebank annotation is ever used for training.

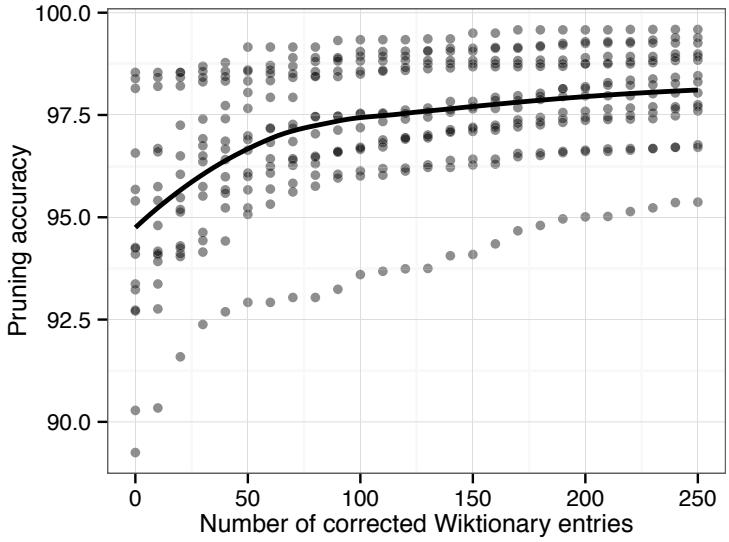

Figure 10.5. Average pruning accuracy (line) across languages (dots) as a function of the number of (hypothetically) corrected Wiktionary entries for the most frequent word types. E.g., position 100 on the x-axis corresponds to manually correcting the entries for the 100 most frequent types. Position 0 corresponds to experimental conditions.

Influence of token and type constraints

The empirical success of the model trained with coupled token and type constraints confirms that these constraints indeed provide complementary signals. Figure 10.4 provides a more detailed view of the relative benefits of each type of constraint. We observe several interesting trends. First, word types that occur with more token constraints during training are generally tagged more accurately, regardless of whether these types occur in Wiktionary. The most common scenario is for a word type to have exactly one tag in Wiktionary and to occur with this projected tag over 100 times in the training set (facet 1, rightmost box). These common word types are typically tagged very accurately across all languages. Second, the word types that are ambiguous according to Wiktionary (facets 2 and 3) are predominantly frequent ones. The accuracy is typically lower for these words compared to the unambiguous words. However, as the number of projected token constraints is increased from zero to 100+ observations, the ambiguous words are effectively disambiguated by the token constraints. This shows the advantage of intersecting token and type constraints. Furthermore, projection generally helps for words that are not in Wiktionary, although the accuracy for these words never reach the accuracy of the words with only one tag in Wiktionary.

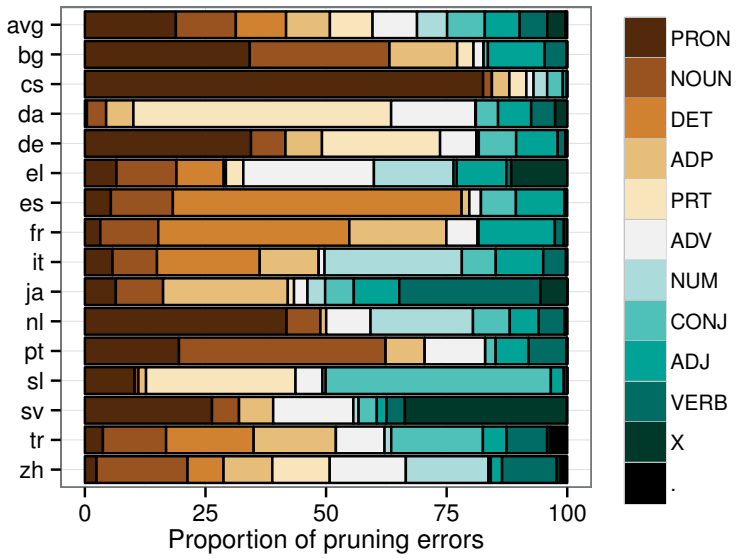

Figure 10.6. Prevalence of pruning mistakes per part-of-speech tag, when pruning the inference search space with Wiktionary.

Wiktionary pruning mistakes

The error analysis by Li et al. (2012) showed that the tags licensed by Wiktionary are often valid. When using Wiktionary to prune the search space of our constrained models and to filter token-level projections, it is also important that correct tags are not mistakenly pruned because they are missing from Wiktionary. While the accuracy of filtering is more difficult to study, due to the lack of a gold standard tagging of the bitext, pruning errors are easily studied by looking at the target language treebank. Figure 10.5 (position 0 on the x-axis) shows that search space pruning errors are not a major issue for most languages; on average the pruning accuracy is almost 95%. However, for some languages such as Chinese and Czech the correct tag is pruned from the search space for nearly 10% of all tokens. When using Wiktionary as a pruner, the upper bound on accuracy for these languages is therefore only around 90%. However, fig. 10.5 also shows that with some manual effort we might be able to remedy many of these errors. For example, by adding missing valid tags to the 250 most common word types in the worst language, the minimum pruning accuracy would rise above 95% from below 90%. If the same was to be done for all of the studied languages, the mean pruning accuracy would reach over 97%.

Figure 10.6 breaks down pruning errors resulting from incorrect or incomplete Wiktionary entries across the correct part-of-speech tags. From this we observe that, for many languages, the pruning errors are highly skewed towards specific tags. For example, for Czech over 80% of the pruning errors are caused by mistakenly pruned pronouns. This suggests that some of these er-

rors may be treebank specific or caused by mistakes in the language specific mappings from the fine-grained tags, used in the treebanks, to the universal tags predicted by our models (see section 10.3.1).

Part IV:
Conclusion

11. Conclusion

The statistical revolution in natural language processing has allowed rapid development of linguistic processing tools that are often more accurate, compared to the manually crafted tools of yore. However, the reliance on fully labeled training data still hinders the widespread availability of these tools for less privileged languages, as well as restricting the accuracy of these tools on non-traditional domains. In this dissertation, we have therefore explored to what extent we can instead train these tools with incomplete and cross-lingual supervision. An overarching theme of this work has been the use of efficient and effective structured discriminative latent variable models.

In summary, we have established that this class of models can be used to create substantially more accurate tools, compared to both unsupervised methods and to recently proposed cross-lingual methods, provided that care is taken both with respect to the model structure and the type of supervision used. The empirical support for this claim is particularly strong in the cross-lingual learning setting; at the time of writing, the cross-lingual models for syntactic dependency parsing and part-of-speech tagging, described in chapters 9 and 10, correspond to the hitherto best published results in the setting where no annotated training data is available in the target language, as evaluated on standardized data sets for a wide variety of languages.

In this chapter, we first look back at our main contributions to the study of linguistic structure prediction with incomplete and cross-lingual supervision; we then outline what we consider to be the most pertinent directions for future research. In addition to further research into more effective models and richer cross-lingual feature representations, we believe that there is significant understanding still to be gained of this learning setting.

11.1 Summary and Main Contributions

Part I (chapters 2 to 4) provides an introduction to linguistic structure and the automatic prediction of such structure from natural language text by means of supervised statistical machine learning. Part II concerns structured prediction with no or incomplete supervision, of which an overview is provided in chapter 5. Finally, Part III is focused on cross-lingual learning, with chapter 7 providing an overview of the major approaches employed in this setting.

The main contributions of this dissertation are found in Part II (chapter 6) and in Part III (chapters 8 to 10). The first contribution is a discriminative latent variable model for fine-grained sentiment analysis with coarse-grained

supervision (chapter 6). The second is a model for cross-lingual word cluster induction and the application thereof to cross-lingual model transfer (chapter 8). The third is a method for adapting multi-source discriminative cross-lingual transfer models to target languages, by means of typologically informed selective parameter sharing (chapter 9). The fourth is an ambiguity-aware self- and ensemble-training algorithm, which is applied to target language adaptation and relexicalization of delexicalized cross-lingual transfer parsers (chapter 9). The fifth is a set of sequence-labeling models which combine constraints at the level of tokens and types, and an instantiation of these models for cross-lingual part-of-speech tagging (chapter 10).

Each of the methods contributed in chapters 6 and 8 to 10 are accompanied by a thorough empirical evaluation and analysis. Finally, a publicly available data set of product reviews, manually annotated at the sentence-level, is provided as part of the study on fine-grained sentiment analysis.

Learning with incomplete supervision

Building on the preliminaries in chapters 3 and 4, where the focus is on supervised approaches to linguistic structure prediction, chapter 5 presents an attempt at a typology of learning scenarios in which the restrictive assumption of full supervision is relaxed. Many of these scenarios can be instantiated by means of structured latent variable models and consequently most of the chapter is devoted to a detailed description of this class of models. In particular, the latter half of the chapter describes how such models can be used for learning to predict linguistic structure in the unsupervised, partially supervised and semi-supervised settings. In contrast to their generative counterparts, where all features need to decompose strictly with the model factorization, discriminative latent variable models admit the use of rich feature representations that decompose arbitrarily over the input.[1]

Sentence-level sentiment analysis with document-level supervision

Chapter 6 considers the use of indirect supervision, in the form of document-level product ratings, to induce sentence-level sentiment. The latter is treated as a latent structure which is placed between the input text and the document-level observations in a hidden conditional random field. We are not the first to propose the use of structured discriminative models with latent variables for learning to make fine-grained predictions with coarse-grained supervision. However, albeit straightforward, the application of such models to sentiment analysis is novel and we are among the first to explicitly consider using the fine-grained latent variable analysis itself for prediction.[2]

[1] Decomposition with respect to the output is restricted even in discriminative models, but they are typically more flexible compared to their generative counterparts in this respect as well).

[2] This work was performed independently of the closely related work of Yessenalina et al. (2010) and Chang et al. (2010).

In order to verify the viability of this approach empirically, a data set of product reviews manually annotated with sentence-level sentiment was produced. This data set is made publicly available and has since its release been used by other researchers, for example, Qu et al. (2012). Our empirical study shows that while sentence-level sentiment can to some degree be induced successfully from document-level sentiment alone, including a small amount of additional data labeled at both levels of granularity results in a substantial performance boost. Importantly, these results suggest that the addition of larger amounts of both types of data lead to further improvements. This result is encouraging, as training the model on millions of product reviews should be quite feasible on commodity hardware with standard parallel optimization methods, such as those discussed in section 4.2.1.

Learning with cross-lingual supervision
Following the study of learning with incomplete monolingual supervision, the remainder of the dissertation is devoted to the multilingual setting, in particular to learning with cross-lingual supervision. The same general class of structured latent variable models is used in both settings. Chapter 7 sets the stage for subsequent chapters by discussing multilingual structure prediction in general and by providing a systematic overview of different scenarios and approaches to learning with cross-lingual supervision in particular. A common motivation for many of these approaches is the tendency of different languages to be ambiguous with respect to different linguistic constructions, which means that the linguistic structure in one language can help disambiguate that in another language. This argument is most commonly made in the scenario where full supervision is available in all languages and in the scenario where no supervision is available in any language.

We argue that the most promising — and at the same time the most plausible — scenario is that where supervision in a subset of languages is used to guide the prediction of linguistic structure in a set of resource-poor languages. In this setting, the key assumption is rather the universality of many linguistic phenomena. This assumption of universality underlies the two key approaches to cross-lingual transfer: annotation projection and model transfer. In the former, annotation in a resource-rich source language is projected to a resource-poor language, typically via word aligned bitext. In the latter, a model is trained to predict the linguistic structure of a resource-rich language, using only cross-lingual features, whereafter the model is applied directly to a resource-poor target language. As discussed in chapter 7, both of these approaches have their shortcomings, some of which are addressed in subsequent chapters.

Cross-lingual word clusters for model transfer
Several studies have shown that coarse-grained non-lexical features carry information which is useful for cross-lingual model transfer for syntactic de-

pendency parsing. However, much information is lost when moving from a fully lexicalized model to a model based solely on coarse-grained part-of-speech tags. It is therefore natural to consider whether more informative cross-lingual features can be induced and leveraged for model transfer.

In chapter 8, we propose to complement the "universal" part-of-speech tags used in prior work with features defined with respect to cross-lingual word clusters. A cross-lingual word clustering is loosely defined as a grouping of words in two languages, such that the groups are consistent across both languages. We provide a simple and efficient algorithm for inducing such clusters from large amounts of monolingual source and target language text together with a smaller amount of word-aligned bitext. The resulting word clusters can readily be incorporated in any transfer model alongside other non-lexical features. In addition to cross-lingual syntactic dependency parsing, we test the viability of these features for cross-lingual named-entity recognition, a task which, to our knowledge, has not been previously considered for model transfer. We show empirically that the inclusion of cross-lingual word clusters yields significantly higher accuracy for both tasks, compared to the delexicalized baseline models; in many cases the improvement is substantial.

In addition to these contributions, this chapter presents an extensive empirical study of the usefulness of monolingual word clusters for parsing and named-entity recognition in the fully supervised monolingual setting. The results of this study confirm that features derived from word clusters provide consistent improvements for both tasks across language families and that these features are particularly useful for named-entity recognition. Given previous studies of monolingual word cluster features, these results are not too surprising. However, this is the first study to consider such a broad scope in terms of both tasks and languages. Based on these results, it is clear that practitioners should, by default, use word cluster features in any model that already uses lexical and part-of-speech features.

Target language adaptation of discriminative transfer parsers
Cross-lingual word cluster features rely on the availability of word-aligned bitext with a resource-rich source language. While such bitext can be found for many of the world's major languages, it may not be available for every target language of interest.[3] Another way to gain improvement over the basic single-source delexicalized transfer model is to combine multiple resource-rich source languages. Most previous approaches to multi-source model transfer for dependency parsing have assumed that the target language model can be expressed as a linear combination of source language model

[3]Moreover, when considering linguistic structure involving multiple lexical items, such as syntactic dependency parsing, the direct correspondence assumption holds to an even lesser extent, which reduces the potential usefulness of annotation projection methods for such structure.

parameters. This assumption generally holds for closely related languages. However, it does not hold in general, as languages often share different subsets of typological traits with different groups of languages. For example, whereas Arabic is similar to both Indo-European and to Romance languages in that they are all prepositional, it differs from Indo-European languages in that adjectives are placed after their head noun; finally, it differs from both with respect to the order of subject, object and verb; see table 9.2.

In chapter 9, we follow recent work by Naseem et al. (2012) and propose to use typological features to selectively share subsets of model features with different source languages. Our contribution to this setting is the incorporation of selective parameter sharing in a discriminative graph-based parser, whereas previous work has been restricted to generative models with strong independence assumptions. In order to accomplish this, some care needs to be taken when combining model features and typological features, but the resulting model is conceptually simple and retains the efficiency of the graph-based model on which it is based; learning and inference can be performed with standard algorithms. We show empirically that the discriminative model with selectively shared features is substantially more accurate in comparison to its generative counterpart, with the exception of a few of the evaluated languages.

Although the model is to some extent adapted to the target language via the typological features, all model features are still solely based on coarse-grained "universal" part-of-speech tags. Previous work has suggested that self-training, in which the transfer model is retrained on its own predictions on target language text, may be useful for further adapting the model to the characteristics of the target language. We argue that standard Viterbi self-training is inappropriate, because too much faith is placed on the model's own predictions. Instead, we introduce an ambiguity-aware self-training algorithm, in which some of the uncertainty of the transfer model is maintained in the form of an ambiguous self-labeling. A fully lexicalized discriminative latent-variable parser is subsequently trained with the generated ambiguous supervision, marginalizing over the preserved uncertainty. This method is flexible; in addition to self-training, we show how it can be used to combine ambiguous predictions from multiple sources in an ensemble model. Empirically, ambiguity-aware self-training and ensemble training both yield significant and consistent improvements over the delexicalized discriminative transfer model with selective parameter sharing. This is in contrast to regular Viterbi self-training, which is not robust across languages and only obtains a negligible improvement on average.

Token and type constraints for cross-lingual part-of-speech tagging

Most approaches to annotation projection rely on the assumption of direct correspondence between the linguistic structures of two languages; an assumption which rarely holds in practice. Furthermore, errors in source-side

predictions and automatic word alignments render the projected annotation both incomplete and noisy. Prior work has shown that filtering the projected annotations is crucial to reducing this noise. However, the improved precision of the projected annotation generally comes at the expense of recall.

Chapter 10 presents a solution to this issue, based on a combination of noisy cross-lingual token-level supervision and ambiguous type-level supervision. The latter is used to filter the transferred token-level annotation. Both levels of constraints are then combined and encoded as an ambiguous labeling, which can subsequently be used to train sequential latent variable models. In particular, the strength of the derived constraints is sufficient for training discriminative models, whereas prior work in this setting has been restricted to generative models. A detailed empirical study shows that this is indeed a successful strategy. Averaged over a variety of languages, we observe a relative error reduction of 25 percent, compared to the prior state of the art. The improvement is consistent across the majority of the languages considered in the evaluation. Additionally, it is shown that monolingual word clusters can in many cases provide a substantial boost in accuracy in this setting. Considered together with the results in chapter 8, this further corroborates the wide applicability of word cluster features for linguistic structure prediction. In fact, even the purely type-supervised model using Wiktionary and monolingual word clusters substantially outperforms the previous state-of-the-art in this setting.

11.2 Future Directions

We have considered a number of different approaches to linguistic structure prediction with incomplete and cross-lingual supervision. Yet, much uncharted territory remains to be explored in this area. We end the dissertation by outlining some interesting and promising directions for future research.

Indirect supervision for sentiment analysis

Despite its simplicity, our model for learning sentence-level sentiment predictions from indirect document-level supervision is quite successful. Nevertheless, there are several ways in which this model could potentially be improved. One possibility that may be worth considering is to include additional constraints on the sentence-level sentiment distributions, using techniques such as posterior regularization (Ganchev et al., 2010). For example, we can expect a positive/negative review to have a certain fraction of positive/negative sentences. The empirical study in chapter 6 suggests that the model trained solely with document-level supervision fails to properly capture this property; the average distribution of sentence-level sentiment diverges considerably from that observed in the test set. We hypothesize that constraining the model's distribution of sentence-level sentiment to be close to this observed

distribution — which could be estimated from a small amount of labeled data — might push the model into a better local optimum during learning. Polarity lexica are another natural source of constraints on sentence-level distributions, as explored by He (2011) in the context of generative topic models.

One could also consider adding additional levels of latent variables to the simple model. For example, one may consider adding latent word-level variables and induce a polarity lexicon jointly with the document- and sentence-level model. If a polarity lexicon is available, the word-level variables could be fixed (or just initialized) to the polarity listed in the lexicon. An even more ambitious route would be to include syntax or discourse into the model, similar to what Nakagawa et al. (2010) proposed for fully supervised sentence-level sentiment analysis. Finally, one could model syntax- or discourse-like structure as latent variables, similar to how such latent structure was encoded in the question-answering model of Liang et al. (2011).

Cross-lingual features

We have shown the usefulness of cross-lingual word cluster features for model transfer in both syntactic dependency parsing and named-entity recognition. Recently, Klementiev et al. (2012) showed that cross-lingual distributed word representations can similarly be used for transfer of text classification models. This opens up the question whether cross-lingual clusters or distributed representations (embeddings) are preferable. The upshot of clusters is their simplicity and efficiency when used as features, since each word is typically assigned to exactly one cluster, which results in a sparse feature representation. This sparseness is also a potential curse, as the clustering can only capture a single salient aspect of each word, whereas embeddings can potentially capture several different aspects. An alternative to embeddings may be to use multiple clusterings; this scenario was recently investigated by Razavi (2012) for syntactic dependency parsing. For further discussion on the use of clusters versus embeddings, see Turian et al. (2010).

Another open question is what type of elements to cluster or embed for different tasks. For example, one could consider inducing cross-lingual clusters of phrases rather than of words. Monolingual phrase clusters have previously been shown to be useful for named-entity recognition (Lin and Wu, 2009), but have not yet been considered in the cross-lingual setting. Klementiev et al. (2012) also point towards the possibility of using cross-lingual distributed representations over phrases; such embeddings were considered in the monolingual setting by Socher et al. (2011). A similar potential direction is to extend the marginalized denoizing autoencoders of Chen et al. (2012) to the cross-lingual setting and to elements other than words.

One of the primary benefits of model transfer with "universal" part-of-speech tags is that no bitext is required. Perhaps the most important question with regard to cross-lingual feature representations is therefore whether it is possible to induce such representations without bitext. Potential points of

departure towards such representations could be kernelized sorting, applied to comparable corpora by Jagarlamudi et al. (2010) or the "translation as decipherment" model proposed by Ravi and Knight (2011) for machine translation without bitext.

Selective parameter sharing

We showed in chapter 9 that resource-rich source languages can be combined for transfer of graph-based dependency parsers via typologically informed selective parameter sharing. One could of course consider this approach for other types of linguistic structure as well. However, it is not clear whether the typological features employed in this work, which specifically capture syntactic characteristics, can be successfully used for other types of linguistic structure. It seems reasonable that these features would be useful for part-of-speech tagging as well, but we are not aware of any attempts in this direction.

Within the context of selective sharing for syntactic parsing, we have thus far only considered a small number of typological features and quite simple ways of selectively sharing model features. More research is needed on which typological features to use and how to encode the selective sharing using these features. In this work only a small number of model features were selectively shared based on typological features; the majority of model features — those encoding word order-dependent local context — were shared based on a division of languages according to a simple notion of familiarity. More refined ways of sharing these features could be considered, such as phylogenetic hierarchies as previously explored by Berg-Kirkpatrick and Klein (2010) for unsupervised dependency parsing. A more complicated approach would be to encode the sharing dynamically, such that the features for each sentence is shared selectively, based on the likelihood that the sentence shares certain typological traits with the different source languages. Søgaard (2011) and Søgaard and Wulff (2012) recently explored such ideas. However, rather than dynamic selective sharing, they performed a dynamically weighted linear interpolation of all source model parameters, which for reasons discussed in chapter 7 cannot work well for typologically diverse languages.

The usefulness of linguistically motivated typological features seems clear. Another option would be to treat these features as latent variables. This was previously considered by Naseem et al. (2012) in their generative model, but could also be incorporated in our discriminative model. The scenarios compared by Naseem et al. were those in which all typological features are observed and in which no typological features are observed. A more promising route may be to treat the typological features as observed, but to add latent variables which encode additional potential typological relationships. Latent variables could also be used as a way to dynamically encode groupings of languages according to family, either in terms of a flat structure or hierarchically as discussed above.

Annotation projection

There are a number of potential avenues for future research on annotation projection. First, in cases where the direct correspondence assumption holds to a sufficient degree, improved alignment models are likely to give direct boosts in accuracy of the projected annotation (Yarowsky and Ngai, 2001). This is already an active area of research within the field of statistical machine translation. The word alignments used in this dissertation were induced with the state-of-the-art alignment model of DeNero and Macherey (2011); while we have not investigated the impact of word alignment errors, we hypothesize that improvements in word alignment will in general translate to improved annotation projection.

In addition to more accurate alignments, we believe that better calibration of the marginal alignment distributions would be useful for annotation projection. Currently, annotation is typically transferred only via high-confident bidirectional alignments, which results in improved alignment precision at a substantial cost of recall. A better approach should be to maintain all the uncertainty in the alignments during the transfer and subsequently marginalize out this uncertainty when training the model on the projected annotation. We considered this approach in preliminary work; however, we observed that the entropy of the marginal alignment distributions were extremely low, so that the distribution contained very little information above that of the Viterbi alignments. With properly calibrated alignment distributions, it would be possible to properly take the uncertainty of the alignments into consideration when projecting the annotation. In addition to using alignment marginals, when a probabilistic model is used to annotate the source language, one could consider transferring the marginal distributions over substructures rather than Viterbi predictions. Thereby, all the uncertainty in the projected annotation would be available for marginalization when training the target model.

Both annotation projection and model transfer have been considered in this dissertation. However, the question of when one of these approaches is preferable to the other remains open. As discussed at length, the direct correspondence assumption used in naïve annotation projection is flawed, in particular when applied to structures involving multiple lexical items. Based on this consideration — and on preliminary unsuccessful attempts at annotation projection for dependency syntax — we chose to use annotation projection for part-of-speech tagging and model transfer for dependency parsing. Still, methods that relax the direct correspondence assumption, such as the quasi-synchronous grammars of Smith and Eisner (2009) and the soft posterior regularization of Ganchev et al. (2009), have been quite successful for transfer of syntactic structure. Although the results on multi-source model transfer with selective sharing in chapter 9 represent the best published so far in this context, annotation projection may still have something to offer for cross-lingual learning of syntax. An alternative to choosing between these

methods may be to simply combine them, for example, via ambiguity-aware ensemble training or via posterior regularization.

11.3 Final Remarks

In conclusion, we have provided constructive affirmative answers to both of the research questions posed in chapter 1 and confirmed our thesis that incomplete and cross-lingual supervision can effectively be leveraged to predict linguistic structure of different types in a wide range of languages, by means of simple and efficient discriminative latent variable models. While results are still substantially below those attainable with full supervision, the models proposed in this dissertation outperform unsupervised approaches with a wide margin. We argue that, for any linguistic processing task of real value, some amount of fully or incompletely labeled data is likely to be available at least in some languages, while creating such supervision for all the world's languages is infeasible. Hence, in addition to its academic merits, this dissertation should also be of value to the natural language processing practitioner.

References

Abeillé, A. and N. Barrier (2004). Enriching a french treebank. In *Proceedings of LREC*. [128]

Abeillé, A., L. Clément, and F. Toussenel (2003). Building a Treebank for French. In A. Abeillé (Ed.), *Treebanks: Building and Using Parsed Corpora*, Chapter 10, pp. 165–187. Springer. [166]

Aberdeen, J., J. Burger, D. Day, L. Hirschman, P. Robinson, and M. Vilain (1995). MITRE: description of the Alembic system used for MUC-6. In *Proceedings of MUC-6*. [23]

Abney, S. (2004). Understanding the yarowsky algorithm. *Computational Linguistics 30*(3), 365–395. [65]

Adesam, Y. (2012). *The Multilingual Forest: Investigating High-quality Parallel Corpus Development*. Ph. D. thesis, Stockholm University, Department of Linguistics, Stockholm, Sweden. [119]

Agarwal, A., O. Chapelle, M. Dudík, and J. Langford (2011). A reliable effective terascale linear learning system. *arXiv:1110.4198v2 [cs.LG]*. [54]

Agarwal, A. and H. Daumé (2009). Exponential family hybrid semi-supervised learning. In *Proceedings of IJCAI*. [102]

Aho, A. V. and J. D. Ullman (1972). *The theory of parsing, translation, and compiling*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc. [118]

Al-Kadi, I. A. (1992). Origins of cryptology: The Arab contributions. *Cryptologica 16*(2), 92–126. [2]

Altun, Y., I. Tsochantaridis, and T. Hofmann (2003). Hidden Markov support vector machines. In *Proceedings of ICML*. [52, 60]

Andersen, P. M., P. J. Hayes, A. K. Heuttner, L. M. Schmandt, and I. B. Nirenberg (1986). Automatic extraction. In *Proceedings of AAAI*. [22]

Andrew, G. and J. Gao (2007). Scalable training of l_1-regularized log-linear models. In *Proceedings of ICML*. [56]

Apresjan, J., I. Boguslavsky, B. Iomdin, L. Iomdin, A. Sannikov, and V. Sizov (2006). A syntactically and semantically tagged corpus of Russian: State of the art and prospects. In *Proceedings of LREC*. [128]

Attardi, G. (2006). Experiments with a multilanguage non-projective dependency parser. In *Proceedings of CoNLL-X*. [20]

Babych, B. and A. Hartley (2003). Improving machine translation quality with automatic named entity recognition. In *Proceedings of the EAMT workshop on Improving MT through other Language Technology Tools*. [21]

Bai, X., R. Padman, and E. Airoldi (2005). On learning parsimonious models for extracting consumer opinions. In *Proceedings of HICSS*. [24, 25]

Baker, J. K. (1979). Trainable grammars for speech recognition. In *Proceedings of the Spring Conference of the Acoustical Society of America*. [44]

Bakır, G., T. Hofmann, B. Schölkopf, A. J. Smola, B. Taskar, and S. Vishwanathan (Eds.) (2007). *Predicting Structured Data*. Neural Information Processing Series. Cambridge, MA, USA: The MIT Press. [29]

Ballesteros, M. and J. Nivre (2012). MaltOptimizer: A system for MaltParser optimization. In *Proceedings of LREC*. [110]

Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation of probabilistic functions of a Markov process. *Inequalities 3*, 1–8. [44]

Beineke, P., T. Hastie, C. Manning, and S. Vaithyanathan (2003). An exploration of sentiment summarization. In *Proceedings of AAAI*. [26]

Belkin, M., P. Niyogi, and V. Sindhwani (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research 7*, 2399–2434. [65]

Bellare, K., G. Druck, and A. McCallum (2009). Alternating projections for learning with expectation constraints. In *Proceedings of UAI*. [68]

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press. [43]

Bender, E. M. (2011). On achieving and evaluating language-independence in NLP. *Linguistic Issues in Language Technology 6*(3), 1–26. [107, 110]

Bennett, S. W., C. Aone, and C. Lovell (1997). Learning to tag multilingual texts through observation. In *Proceedings of EMNLP*. [23]

Berg-Kirkpatrick, T., A. Bouchard-Côté, J. DeNero, and D. Klein (2010). Painless unsupervised learning with features. In *Proceedings of NAACL-HLT*. [42, 74, 75, 159]

Berg-Kirkpatrick, T. and D. Klein (2010). Phylogenetic grammar induction. In *Proceedings of ACL*. [108, 111, 120, 138, 141, 142, 149, 186]

Berger, A., V. Della Pietra, and S. Della Pietra (1996). A maximum entropy approach to natural language processing. *Computational Linguistics 22*(1), 39–71. [14, 20, 40]

Bethard, S., H. Yu, A. Thornton, V. Hatzivassiloglou, and D. Jurafsky (2004). Automatic extraction of opinion propositions and their holders. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text*. [24]

Bikel, D. M., R. Schwartz, and R. M. Weischedel (1999). An algorithm that learns what's in a name. *Machine Learning 34*(1), 211–231. [23]

Bisang, W. (2010). Word classes. In J. J. Song (Ed.), *The Oxford handbook of Language Typology*. Oxford: Oxford University Press. [13]

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer New York. [66]

Blair-Goldensohn, S., K. Hannan, R. McDonald, T. Neylon, G. A. Reis, and J. Reynar (2008). Building a sentiment summarizer for local service reviews. In *Proceedings of the WWW Workshop on NLP in the Information Explosion Era (NLPIX)*. [26, 82]

Blitzer, J., M. Dredze, and F. Pereira (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of ACL*. [26, 94]

Blitzer, J., R. McDonald, and F. Pereira (2006). Domain adaptation with structural correspondence learning. In *Proceedings of EMNLP*. [14]

Blum, A. and T. Mitchell (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of COLT*. [65]

Boguslavsky, I., S. Grigorieva, N. Grigoriev, L. Kreidlin, and N. Frid (2000). Dependency treebank for Russian: Concept, tools, types of information. In *Proceedings of COLING*. [128]

Böhmová, A., J. Hajič, E. Hajičová, and B. Hladká (2003). The PDT: a 3-level annotation scenario. In A. Abeillé (Ed.), *Treebanks: Building and Using Parsed Corpora*, Chapter 7, pp. 103–127. Springer. [122]

Bottou, L. (2004). Stochastic learning. In O. Bousquet and U. von Luxburg (Eds.), *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, pp. 146–168. Berlin: Springer Verlag. [57]

Bottou, L. and O. Bousquet (2007). The tradeoffs of large scale learning. In *Proceedings of NIPS*. [54, 57]

Bouma, G., J. Kuhn, B. Schrader, and K. Spreyer (2008). Parallel LFG grammars on parallel corpora: A base for practical triangulation. In *Proceedings of LFG*. [115, 117, 120]

Bousquet, O., O. Chapelle, and M. Hein (2004). Measure based regularization. In *Proceedings of NIPS*. [65]

Boyd, S. and L. Vandenberghe (2004). *Convex Optimization*. New York, NY, USA: Cambridge University Press. [49, 53]

Brants, T. (2000). TnT: a statistical part-of-speech tagger. In *Proceedings of ANLP*. [14]

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of ANLP*. [14]

Brill, E. (1995). Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging. *Computational Linguistics 21*(4), 543–565. [14, 23]

Brill, E., D. Magerman, M. Marcus, and B. Santorini (1990). Deducing linguistic structure from the statistics of large corpora. In *Proceedings of the DARPA Speech and Natural Language Workshop*. [159]

Brill, E. and M. Marcus (1992). Tagging an unfamiliar text with minimal human supervision. In *Proceedings of the AAAI Fall Symposium: Probabilistic Approaches to Natural Language*. [66, 159]

Brown, P. F., P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai (1992). Class-based n-gram models of natural language. *Computational Linguistics 18*, 467–479. [126, 127, 129, 159]

Buch-Kromann, M., I. Korzen, and H. Høeg Müller (2009). Uncovering the "lost" structure of translations with parallel treebanks. *Copenhagen Studies in Language 38*, 199–224. [119]

Buchholz, S. and E. Marsi (2006). CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL*. [107, 110, 128, 166]

Burkett, D., J. Blitzer, and D. Klein (2010). Joint parsing and alignment with weakly synchronized grammars. In *Proceedings of HLT*. [109, 118]

Burkett, D. and D. Klein (2008). Two languages are better than one (for syntactic parsing). In *Proceedings of EMNLP*. [109, 110]

Burkett, D., S. Petrov, J. Blitzer, and D. Klein (2010). Learning better monolingual models with unannotated bilingual text. In *Proceedings of CoNLL*. [110]

Candito, M., B. Crabbé, and P. Denis (2010). Statistical french dependency parsing: treebank conversion and first results. In *Proceedings of LREC*. [128]

Carenini, G., R. Ng, and A. Pauls (2006). Multi-document summarization of evaluative text. In *Proceedings of EACL*. [26]

Carreras, X. (2007). Experiments with a higher-order projective dependency parser. In *Proceedings of CoNLL Shared Task*. [18, 36]

Celeux, G. and G. Govaert (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis 14*(3), 315–332. [65]

Cesa-Bianchi, N. and G. Lugosi (2006). *Prediction, Learning, and Games*. New York, NY, USA: Cambridge University Press. [46, 50]

Chang, M.-W., D. Goldwasser, D. Roth, and V. Srikumar (2010). Discriminative learning over constrained latent representations. In *Proceedings of NAACL*. [69]

Chang, M.-W., L. Ratinov, N. Rizzolo, and D. Roth (2008). Learning and inference with constraints. In *Proceedings of AAAI*. [68]

Chang, M.-W., L. Ratinov, and D. Roth (2007). Guiding semi-supervision with constraint-driven learning. In *Proceedings of ACL*. [68]

Chang, M.-W., L. Ratinov, and D. Roth (2012). Structured learning with constrained conditional models. *Machine Learning 88*(3), 399–431. [68]

Chang, M.-W., V. Srikumar, D. Goldwasser, and D. Roth (2010). Structured output learning with indirect supervision. In *Proceedings of ICML*. [67, 102, 180]

Chapelle, O., B. Schölkopf, and A. Zien (Eds.) (2006). *Semi-Supervised Learning*. Adaptive Computation and Machine Learning. The MIT Press. [64]

Chapelle, O., J. Weston, and B. Schölkopf (2003). Cluster kernels for semi-supervised learning. In *Advances In Neural Information Processing Systems*, Volume 15 of *NIPS*. [65]

Charniak, E. and M. Johnson (2005). Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of ACL*. [34, 150, 151]

Chen, D., C. Dyer, S. B. Cohen, and N. A. Smith (2011). Unsupervised bilingual POS tagging with Markov random fields. In *Proceedings of the First Workshop on Unsupervised Learning in NLP*. [111]

Chen, M., Z. Xu, K. Weinberger, and F. Sha (2012). Marginalized denoising autoencoders for domain adaptation. In *Proceedings of ICML*. [127, 185]

Chen, S. F. (2003). Conditional and joint models for grapheme-to-phoneme conversion. In *Proceedings of Eurospeech*. [42, 74]

Chen, S. F. and R. Rosenfeld (1999). A gaussian prior for smoothing maximum entropy models. Technical report, School of Computer Science, Carnegie Mellon University. [53]

Chen, W., J. Kazama, and K. Torisawa (2010). Bitext dependency parsing with bilingual subtree constraints. In *Proceedings of ACL*. [109, 110]

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation. In *Proceedings of ACL*. [15]

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics 33*(2), 201–228. [19, 109]

Choi, F. Y. Y. (2000). Advances in domain independent linear text segmentation. In *Proceedings of NAACL*. [12]

Choi, Y. and C. Cardie (2009). Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of EMNLP*. [26, 81, 82]

Choi, Y., C. Cardie, E. Riloff, and S. Patwardhan (2005). Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of HLT-EMNLP*. [24]

Christodoulopoulos, C., S. Goldwater, and M. Steedman (2010). Two decades of unsupervised POS induction: How far have we come? In *Proceedings of EMNLP*. [159]

Church, K. W. (1988). A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of ANLC*. [14]

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement 20*(1), 37–46. [80]

Cohen, S. B., D. Das, and N. A. Smith (2011). Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of EMNLP*. [113, 119, 120, 121, 142, 150]

Cohen, S. B. and N. A. Smith (2009). Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of NAACL*. [108, 111, 120, 141]

Collins, M. (1997). Three generative, lexicalised models for statistical parsing. In *Proceedings of ACL*. [17]

Collins, M. (2002). Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In *Proceedings of EMNLP*. [20, 50, 58]

Collins, M., J. Hajič, L. Ramshaw, and C. Tillmann (1999). A statistical parser for Czech. In *Proceedings of ACL*. [18]

Collins, M., P. Koehn, and I. Kučerová (2005). Clause restructuring for statistical machine translation. In *Proceedings of ACL*. [15]

Collins, M. and B. Roark (2004). Incremental parsing with the perceptron algorithm. In *Proceedings of ACL*. [30]

Collins, M. and Y. Singer (1999). Unsupervised models for named entity classification. In *Proceedings of EMNLP-VLC*. [65, 66]

Collobert, R. and J. Weston (2008). A unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML*. [65, 127]

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Journal of Machine Learning Research 20*(3), 273–297. [20, 52]

Councill, I., R. McDonald, and L. Velikovich (2010). What's great and what's not: Learning to classify the scope of negation for improved sentiment analysis. In *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*. [15, 82, 88]

Covington, M. A. (2001). A fundamental algorithm for dependency parsing. In *Proceedings of ACM-SE*. [19]

Cowie, J. and Y. Wilks (2000). Information extraction. In H. M. R. Dale and H. Somers (Eds.), *Handbook of Natural Language Processing*. New York, NY, USA: Marcel Dekker. [21]

Crammer, K., O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer (2006). Online passive-aggressive algorithms. *Journal of Machine Learning Research 7*, 551–585. [56]

Crammer, K. and Y. Singer (2003). Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research 3*, 951–991. [86]

Croft, W. (1991). *Syntactic Categories and Grammatical Relations: The Cognitive Organization of Information.* Chicago, IL, USA: University of Chicago Press. [13]

Daelemans, W. and A. van den Bosch (2005). *Memory-Based Language Processing.* Cambridge, UK: Cambridge University Press. [14, 20]

Daelemans, W., J. Zavrel, P. Berck, and S. Gillis (1996). MBT: A memory-based part of speech tagger-generator. In *Proceedings of VLC.* [14]

Das, D. (2012). *Semi-Supervised and Latent-Variable Models of Natural Language Semantics.* Ph. D. thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA. [17]

Das, D., A. F. T. Martins, and N. A. Smith (2012). An exact dual decomposition algorithm for shallow semantic parsing with constraints. In *Proceedings of *SEM.* [68]

Das, D. and S. Petrov (2011). Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of ACL-HLT.* [42, 108, 113, 118, 137, 158, 159, 161, 166, 167, 168, 169, 171]

Daumé, III, H. and D. Marcu (2005). Learning as search optimization: approximate large margin methods for structured prediction. In *Proceedings of ICML.* [20, 30]

Daumé III, H. (2006). *Practical structured learning techniques for natural language processing.* Ph. D. thesis, University of Southern California, Los Angeles, CA, USA. [29, 30, 57]

Daumé III, H. (2007). Frustratingly easy domain adaptation. In *Proceedings of ACL.* [147]

Daumé III, H., J. Langford, and D. Marcu (2009). Search-based structured prediction. *Journal of Machine Learning Research 75*(3), 297–325. [20, 30]

Dave, K., S. Lawrence, and D. M. Pennock (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of WWW.* [25, 26]

Davison, A. C. and D. V. Hinkley (1997). *Bootstrap Methods and Their Applications.* Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge, UK: Cambridge University Press. [89]

De Marneffe, M.-C., B. MacCartney, and C. D. Manning (2006). Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC.* [128, 137]

de Marneffe, M.-C. and C. D. Manning (2008). Stanford typed dependencies manual. [15, 36, 122]

Dembczynski, K., W. Waegeman, W. Cheng, and E. Hüllermeier (2011). An exact algorithm for F-measure maximization. In *Proceedings of NIPS.* [49]

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B 39*, 1–38. [74]

DeNero, J. and K. Macherey (2011). Model-based aligner combination using dual decomposition. In *Proceedings of ACL-HLT.* [138, 167, 187]

Denis, P. and J. Baldridge (2007, April). Joint determination of anaphoricity and coreference resolution using integer programming. In *Proceedings of NAACL-HLT*. [43]

DeRose, S. J. (1988). Grammatical category disambiguation by statistical optimization. *Computational Linguistics 14*(1), 31–39. [14]

Derouault, A. M. and B. Merialdo (1986, June). Natural language modeling for phoneme-to-text transcription. *IEEE Transactions on Pattern Analysis and Machine Intelligence 8*(6), 742–749. [14]

Dhillon, P., D. Foster, and L. Dean (2011). Multi-view learning of word embeddings via CCA. In *Proceedings of NIPS*. [64, 127]

Dhillon, P., J. Rodu, D. Foster, and L. Ungar (2012). Using CCA to improve CCA: A new spectral method for estimating vector models of words. In *Proceedings of ICML*. [127, 159]

Diab, M. and P. Resnik (2002). An unsupervised method for word sense tagging using parallel corpora. In *Proceedings of ACL*. [113]

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, pp. 1–15. [38]

Dietterich, T. G. (2002). Machine learning for sequential data: A review. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*. [30]

Dredze, M., P. P. Talukdar, and K. Crammer (2009). Sequence learning from data with multiple labels. In *Proceedings of the ECML/PKDD Workshop on Learning from Multi-Label Data*. [150, 151]

Dryer, M. S. and M. Haspelmath (Eds.) (2011). *The World Atlas of Language Structures Online*. Max Planck Digital Library. [13, 121, 142, 147]

Duan, X., J. Zhao, and B. Xu (2007). Probabilistic models for action-based Chinese dependency parsing. In *Proceedings of ECML*. [20]

Duchi, J., E. Hazan, and Y. Singer (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research 12*(July), 2121–2159. [57]

Duchi, J., S. Shalev-Shwartz, Y. Singer, and T. Chandra (2008). Efficient projections onto the l_1-ball for learning in high dimensions. In *Proceedings of ICML*. [56]

Durrett, G., A. Pauls, and D. Klein (2012). Syntactic transfer using a bilingual lexicon. In *Proceedings of EMNLP-CoNLL*. [119]

Efron, B. and R. J. Tibshirani (1993). *An Introduction to the Bootstrap*. New York, NY, USA: Chapman & Hall. [89, 170]

Eisner, J. (1996). Three new probabilistic models for dependency parsing: an exploration. In *Proceedings of COLING*. [18, 19, 35, 44, 145]

Eisner, J. (2002). Parameter estimation for probabilistic finite-state transducers. In *Proceedings of ACL*. [44]

Eisner, J. (2003). Learning non-isomorphic tree mappings for machine translation. In *Proceedings of ACL*. [118]

Ekman, P. (1993). Facial expression and emotion. *American Psychologist 48*, 384–392. [24]

Esuli, A. and F. Sebastiani (2006a). Determining term subjectivity and term orientation for opinion mining. In *Proceedings of EACL*. [103]

Esuli, A. and F. Sebastiani (2006b). SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*. [25]

Faruqui, M. and S. Padó (2010). Training and evaluating a German named entity recognizer with semantic generalization. In *Proceedings of KONVENS*. [126, 131, 132]

Fillmore, C. J. (1982). Frame semantics. In *Linguistics in the Morning Calm*. Seoul, South Korea: Hanshin Publishing Company. [17]

Finch, S. and N. Chater (1992). Bootstrapping syntactic categories using statistical methods. In W. Daelemans and D. Powers (Eds.), *Background and Experiments in Machine Learning of Natural Language*, pp. 229–236. Tilburg, NL: ITK. [159]

Finkel, J. R., A. Kleeman, and C. D. Manning (2008). Efficient, feature-based, conditional random field parsing. In *Proceedings of ACL-HLT*. [18]

Finkel, J. R. and C. D. Manning (2009). Hierarchical Bayesian domain adaptation. In *Proceedings of HLT-NAACL*. [147]

Fossum, V. and S. Abney (2005). Automatically inducing a part-of-speech tagger by projecting from multiple source languages across aligned corpora. In *Proceedings of IJCNLP*. [113, 117, 120]

Fossum, V. and K. Knight (2008). Using bilingual Chinese-English word alignments to resolve PP-attachment ambiguity in English. In *Proceedings of the AMTA Student Workshop*. [109, 110]

Fraser, A., R. Wang, and H. Schütze (2009). Rich bitext projection features for parse reranking. In *Proceedings of EACL*. [110]

Freitag, D. (2004). Trained named entity recognition using distributional clusters. In *Proceedings of EMNLP*. [64, 126]

Freund, Y. and R. E. Schapire (1999). Large margin classification using the Perceptron algorithm. *Journal of Machine Learning Research 37*(3), 277–296. [57, 58]

Fundel, K., R. Küffner, and R. Zimmer (2007). Relex—relation extraction using dependency parse trees. *Bioinformatics 23*(3), 365–371. [15]

Gaifman, H. (1965). Dependency systems and phrase-structure systems. *Information and Control 8*, 304–337. [19]

Gallo, G., G. Longo, S. Pallottino, and S. Nguyen (1993). Directed hypergraphs and applications. *Discrete Applied Mathematics 42*(2-3), 177–201. [44]

Gamon, M., A. Aue, S. Corston-Oliver, and E. Ringger (2005). Pulse: Mining customer opinions from free text. In *Proceedings of IDA*. [26]

Ganchev, K., J. Gillenwater, and B. Taskar (2009). Dependency grammar induction via bitext projection constraints. In *Proceedings of ACL-IJCNLP*. [108, 113, 118, 122, 123, 187]

Ganchev, K., J. Graça, J. Gillenwater, and B. Taskar (2010). Posterior regularization for structured latent variable models. *Journal of Machine Learning Research 11*(July), 2001–2049. [68, 69, 118, 184]

Garrette, D. and J. Baldridge (2012). Type-supervised hidden Markov models for part-of-speech tagging with incomplete tag dictionaries. In *Proceedings of EMNLP-CoNLL*. [158]

Garrette, D. and J. Baldridge (2013). Learning a part-of-speech tagger from two hours of annotation. In *Proceedings of NAACL-HLT*. [158]

Garside, R., G. Sampson, and G. N. Leech (1987). *The Computational analysis of English: a corpus-based approach*. London, UK: Longman. [14]

Gee, J. P. (2011). *An Introduction to Discourse Analysis: Theory and Method* (3rd ed.). London, UK: Taylor & Francis Group. [12]

Gelling, D., T. Cohn, P. Blunsom, and J. a. Graça (2012). The PASCAL challenge on grammar induction. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure (WILS)*. [112, 159]

Ghahramani, Z. (2004). *Unsupervised learning*, pp. 72–112. Advanced Lectures on Machine Learning. Springer-Verlag. [64]

Gillenwater, J., K. Ganchev, J. a. Graça, F. Pereira, and B. Taskar (2010). Sparsity in dependency grammar induction. In *Proceedings of ACL*. [108]

Gimpel, K., D. Das, and N. A. Smith (2010). Distributed asynchronous online learning for natural language processing. In *Proceedings of CoNLL*. [54]

Gimpel, K., N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith (2011). Part-of-speech tagging for Twitter: annotation, features, and experiments. In *Proceedings of ACL-HLT*. [12]

Gimpel, K. and N. A. Smith (2010). Softmax-margin CRFs: training log-linear models with cost functions. In *Proceedings of NAACL-HLT*. [52]

Globerson, A., G. Chechik, F. Pereira, and N. Tishby (2007). Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research 8*, 2265–2295. [127]

Goldberg, A. B. and X. Zhu (2006). Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *Proceedings of TextGraphs*. [25, 26]

Goldberg, Y., M. Adler, and M. Elhadad (2008). EM can find pretty good HMM POS-taggers (when given a good start). In *Proceedings of ACL-HLT*. [159]

Goldwater, S. and M. Johnson (2005). Representational bias in unsupervised learning of syllable structure. In *Proceedings of CoNLL*. [64]

Goodman, J. (1999). Semiring parsing. *Computational Linguistics 25*, 573–605. [44]

Grandvalet, Y. and Y. Bengio (2004). Semi-supervised learning by entropy minimization. In *Proceedings of NIPS*. [65]

Greene, B. B. and G. M. Rubin (1971). Automated grammatical tagging of English. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US. [14]

Grishman, R. and B. Sundheim (1996). Message understanding conference – 6: A brief history. In *Proceedings of COLING*. [22]

Haffari, G., M. Razavi, and A. Sarkar (2011). An ensemble model that combines syntactic and semantic clustering for discriminative dependency parsing. In *Proceedings of ACL*. [126]

Haghighi, A. and D. Klein (2006). Prototype-driven learning for sequence models. In *Proceedings of HLT-NAACL*. [66]

Hall, K., R. McDonald, J. Katz-Brown, and M. Ringgaard (2011). Training dependency parsers by jointly optimizing multiple objectives. In *Proceedings of EMNLP*. [69]

Han, C.-h., N.-R. Han, E.-S. Ko, and M. Palmer (2002). Development and evaluation of a Korean treebank and its application to NLP. In *Proceedings of LREC*. [128]

Haspelmath, M. (2001). Word classes/parts of speech. In N. J. Baltes, Paul B. & Smelser (Ed.), *International Encyclopedia of the Social and Behavioral Sciences*, pp. 16538–16545. Amsterdam, The Netherlands: Pergamon. [13]

Hatzivassiloglou, V. and K. R. McKeown (1997). Predicting the semantic orientation of adjectives. In *Proceedings of EACL*. [26]

Haulrich, M. (2012). *Data-driven Bitext Dependency Parsing and Alignment*. Ph. D. thesis, Copenhagen Business School, Copenhagen, Denmark. [110, 119]

Haussler, D. (1992). Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation 100*(1), 78–150. [47]

Hays, D. G. (1964). Dependency theory: A formalism and some observations. *Language* (40), 511–525. [19]

Hazan, E., A. Agarwal, and S. Kale (2007). Logarithmic regret algorithms for online convex optimization. *Journal of Machine Learning Research 69*(December), 169–192. [56]

Hazan, T. and R. Urtasun (2010). A primal-dual message-passing algorithm for approximated large scale structured prediction. In *Proceedings of NIPS*. [52, 72]

He, Y. (2011). Latent sentiment model for weakly-supervised cross-lingual sentiment classification. In *Proceedings of ECIR*. [102, 185]

Headden III, W. P., D. McClosky, and E. Charniak (2008). Evaluating unsupervised part-of-speech tagging for grammar induction. In *Proceedings of COLING*. [112]

Hoerl, A. E. and R. W. Kennard (1970). Ridge regression: Applications to nonorthogonal problems. *Technometrics 12*(1), 69–82. [53]

Holmes, J. and W. Holmes (2002). *Speech Synthesis and Recognition*. Aspects of information technology. London, UK: Taylor & Francis Group. [11]

Hu, M. and B. Liu (2004a). Mining and summarizing customer reviews. In *Proceedings of KDD*. [25, 26, 89, 102]

Hu, M. and B. Liu (2004b). Mining opinion features in customer reviews. In *Proceedings of AAAI*. [25]

Huang, L. (2008a). Advanced dynamic programming in semiring and hypergraph frameworks. COLING tutorial notes. [44]

Huang, L. (2008b). Forest reranking: Discriminative parsing with non-local features. In *Proceedings of ACL*. [20]

Huang, L., S. Fayong, and Y. Guo (2012a). Structured perceptron with inexact search. In *Proceedings of NAACL-HLT*. [20]

Huang, L., S. Fayong, and Y. Guo (2012b). Structured perceptron with inexact search. In *Proceedings of NAACL-HLT*. [20]

Huang, L., W. Jiang, and Q. Liu (2009). Bilingually-constrained (monolingual) shift-reduce parsing. In *Proceedings of EMNLP*. [110]

Huang, L. and K. Sagae (2010). Dynamic programming for linear-time incremental parsing. In *Proceedings of ACL*. [20]

Hudson, R. (1984). *Word Grammar*. Oxford, UK: Blackwell Publishing. [16, 17]

Hwa, R., P. Resnik, A. Weinberg, C. Cabezas, and O. Kolak (2005). Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering 11*(03), 311–325. [108, 113, 114, 115, 116, 117, 118, 122, 123]

Hwa, R., P. Resnik, A. Weinberg, and O. Kolak (2002). Evaluating translational correspondence using annotation projection. In *Proceedings of ACL*. [115]

Jagarlamudi, J., S. Juarez, and H. D. III (2010). Kernelized sorting for natural language processing. In *Proceedings of AAAI*. [186]

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters 31*(8), 651–666. [64]

Jiang, J., P. Rai, and H. Daumé III (2011). Message-passing for approximate map inference with latent variables. In *Proceedings of NIPS*. [71]

Jiao, F., S. Wang, C.-H. Lee, R. Greiner, and D. Schuurmans (2006). Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *Proceedings of ACL*. [65]

Jin, R. and Z. Ghahramani (2002). Learning with multiple labels. In *Proceedings of NIPS*. [150]

Joachims, T. (2005). A support vector method for multivariate performance measures. In *Proceedings of ICML*. [49]

Johansson, R. (2008, December). *Dependency-based Semantic Analysis of Natural-language Text*. Ph. D. thesis, Department of Computer Science, Lund University, Lund, Sweden. [17]

Johansson, R. and A. Moschitti (2013). Relational features in fine-grained opinion analysis. *Computational Linguistics 39*(3). [24]

Jurafsky, D. and J. H. Martin (2009). *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition* (2nd ed.). Upper Saddle River, NJ, USA: Prentice Hall. [11, 12]

Kahane, S. (1997). Bubble trees and syntactic representations. In *Proceedings of MOL*. [16]

Kaji, N. and M. Kitsuregawa (2007). Building lexicon for sentiment analysis from massive collection of HTML documents. In *Proceedings of EMNLP-CoNLL*. [26]

Källgren, G. (1996). Linguistic indeterminacy as a source of errors in tagging. In *Proceedings of COLING*. [14]

Katz-Brown, J., S. Petrov, R. McDonald, F. Och, D. Talbot, H. Ichikawa, M. Seno, and H. Kazawa (2011). Training a parser for machine translation reordering. In *Proceedings of EMNLP*. [15, 69]

Kim, S., K. Toutanova, and H. Yu (2012). Multilingual named entity recognition using parallel data and metadata from Wikipedia. In *Proceedings of ACL*. [108]

Kim, S.-M. and E. Hovy (2004). Determining the sentiment of opinions. In *Proceedings of COLING*. [26, 103]

Kim, S.-M. and E. Hovy (2006a). Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of the ACL/COLING Workshop on Sentiment and Subjectivity in Text*. [24]

Kim, S.-M. and E. Hovy (2006b). Identifying and analyzing judgment opinions. In *Proceedings of HLT-NAACL*. [24, 25]

Kindermann, R. and J. L. Snell (1980). *Markov Random Fields and Their Applications*. American Mathematical Society. [41]

Kivinen, J., A. J. Smola, and R. C. Williamson (2004). Online learning with kernels. *IEEE Transactions on Signal Processing 52*(8), 2165—2176. [57]

Kivinen, J. and M. K. Warmuth (1997). Exponentiated gradient versus gradient descent for linear predictors. *Journal Information and Computation 132*(1), 1–63. [56]

Klein, D. (2005). *The unsupervised learning of natural language structure.* Ph. D. thesis, Stanford University, Stanford, CA, USA. [112]

Klein, D. and C. D. Manning (2001). Parsing and hypergraphs. In *Proceedings of IWPT.* [44]

Klein, D. and C. D. Manning (2002). A generative constituent-context model for improved grammar induction. In *Proceedings of ACL.* [112]

Klein, D. and C. D. Manning (2004). Corpus-based induction of syntactic structure: models of dependency and constituency. In *Proceedings of ACL.* [108, 112]

Klementiev, A., I. Titov, and B. Bhattarai (2012). Inducing crosslingual distributed representations of words. In *Proceedings of COLING.* [119, 185]

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT Summit.* [167]

Koller, D. and N. Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques.* Adaptive Computation and Machine Learning. MIT Press. [40, 84]

Koo, T., X. Carreras, and M. Collins (2008). Simple semi-supervised dependency parsing. In *Proceedings of ACL-HLT.* [126, 131]

Koo, T. and M. Collins (2010). Efficient third-order dependency parsers. In *Proceedings of ACL.* [18, 36]

Koo, T., A. M. Rush, M. Collins, T. Jaakkola, and D. Sontag (2010). Dual decomposition for parsing with non-projective head automata. In *Proceedings of EMNLP.* [19]

Kroeger, P. R. (2005). *Analyzing Grammar: An Introduction.* Cambridge, UK: Cambridge University Press. [12, 15]

Kudo, T. and Y. Matsumoto (2000). Japanese dependency structure analysis based on support vector machines. In *Proceedings of EMNLP-VLC.* [20]

Kuhlmann, M. (2013). Mildly non-projective dependency grammar. *Computational Linguistics 39*(2). [17]

Kuhlmann, M., C. Gómez-Rodríguez, and G. Satta (2011). Dynamic programming algorithms for transition-based dependency parsers. In *Proceedings of ACL.* [20]

Kuhn, J. (2004). Experiments in parallel-text based grammar induction. In *Proceedings of ACL.* [111]

Kummerfeld, J. K., D. Hall, J. R. Curran, and D. Klein (2012). Parser showdown at the Wall Street corral: an empirical investigation of error types in parser output. In *Proceedings of EMNLP-CoNLL.* [113]

Kwon, N., L. Zhou, E. Hovy, and S. W. Shulman (2007). Identifying and classifying subjective claims. In *Proceedings of DG.* [24, 25]

Lafferty, J., A. McCallum, and F. Pereira (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML.* [14, 41]

Lamar, M., Y. Maron, and E. Bienenstock (2010). Latent-descriptor clustering for unsupervised POS induction. In *Proceedings of EMNLP.* [127, 159]

Lamar, M., Y. Maron, M. Johnson, and E. Bienenstock (2010). SVD and clustering for unsupervised POS tagging. In *Proceedings of ACL.* [66]

Lappin, S. (1997). *The Handbook of Contemporary Semantic Theory*. Blackwell Handbooks in Linguistics. Oxford, UK: Wiley. [12]

Lasserre, J., C. Bishop, and T. Minka (2006). Principled hybrids of generative and discriminative models. In *Proceedings of CVPR*. [39, 102]

Lee, L. (2004). A matter of opinion: Sentiment analysis and business intelligence. IBM Faculty Summit on the Architecture of On-Demand Business. [23, 25]

Leidner, J. L., G. Sinclair, and B. Webber (2003). Grounding spatial named entities for information extraction and question answering. In *Proceedings of HLT-NAACL-GEOREF*. [21]

Lewis, II, P. M. and R. E. Stearns (1968). Syntax-directed transduction. *Journal of the ACM 15*(3), 465–488. [118]

Li, S., J. a. Graça, and B. Taskar (2012). Wiki-ly supervised part-of-speech tagging. In *Proceedings of EMNLP-CoNLL*. [42, 75, 158, 159, 164, 166, 167, 168, 169, 174]

Li, X., S. Strassel, S. Grimes, S. Ismael, M. Maamouri, A. Bies, and N. Xue (2012). Parallel aligned treebanks at LDC: New challenges interfacing existing infrastructures. In *Proceedings of LREC*. [119]

Liang, P., M. I. Jordan, and D. Klein (2009). Learning from measurements in exponential families. In *Proceedings of ICML*. [68, 69]

Liang, P., M. I. Jordan, and D. Klein (2011). Learning dependency-based compositional semantics. In *Proceedings of ACL*. [185]

Lin, C. and Y. He (2009). Joint sentiment/topic model for sentiment analysis. In *Proceeding of CIKM*. [102]

Lin, C., Y. He, R. Everson, and S. M. Rüger (2012). Weakly supervised joint sentiment-topic detection from text. *IEEE Transactions on Knowledge and Data Engineering 24*(6), 1134–1145. [102]

Lin, D. and X. Wu (2009). Phrase clustering for discriminative learning. In *Proceedings of ACL-IJCNLP*. [127, 131, 185]

Liu, B. (2010). Sentiment analysis and subjectivity. *Handbook of Natural Language Processing 5*(1), 1–38. [78]

Liu, D. C. and J. Nocedal (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming 45*, 503–528. [54, 145]

Liu, H., H. Lieberman, and T. Selker (2003). A model of textual affect sensing using real-world knowledge. In *Proceedings of IUI*. [24]

Liu, Q. and A. Ihler (2011). Variational algorithms for marginal MAP. In *Proceedings of UAI*. [71]

Ma, X. (2010). Toward a name entity aligned bilingual corpus. In *Proceedings of the LREC Acquisition Workshop*. [119]

MacKay, D. J. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press. [3]

Magerman, D. M. (1995). Statistical decision-tree models for parsing. In *Proceedings of ACL*. [17]

Maier, W., E. Hinrichs, S. Kübler, and J. Krivanek (2012). Annotating coordination in the Penn treebank. In *Proceedings of LAW*. [121]

Mann, G., R. Mcdonald, M. Mohri, N. Silberman, and D. D. Walker (2009). Efficient large-scale distributed training of conditional maximum entropy models. In

Proceeding of NIPS. [54]

Mann, G. S. and A. McCallum (2008). Generalized expectation criteria for semi-supervised learning of conditional random fields. In *Proceedings of ACL*. [69]

Manning, C. D. (2011). Part-of-speech tagging from 97% to 100%: is it time for some linguistics? In *Proceedings of CICLing*, pp. 171–189. [14]

Mao, Y. and G. Lebanon (2006). Isotonic conditional random fields and local sentiment flow. In *Proceedings of NIPS*. [26]

Maosong, S., S. Dayang, and B. K. Tsou (1998). Chinese word segmentation without using lexicon and hand-crafted training data. In *Proceedings of COLING*. [12]

Marcu, D. (2000). *The Theory and Practice of Discourse Parsing and Summarization*. Cambridge, MA, USA: MIT Press. [12]

Marcus, M. P., M. A. Marcinkiewicz, and B. Santorini (1993). Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics 19*(2), 313–330. [128, 167]

Maron, Y., M. Lamar, and E. Bienenstock (2010). Sphere embedding: An application to part-of-speech induction. In *Proceedings of NIPS*. [127, 159]

Martins, A. F. T., N. A. Smith, P. M. Q. Aguiar, and M. A. T. Figueiredo (2011). Dual decomposition with many overlapping components. In *Proceedings of EMNLP*. [19]

Martins, A. F. T., N. A. Smith, and E. P. Xing (2009). Concise integer linear programming formulations for dependency parsing. In *Proceedings of ACL*. [18, 19, 43]

Martins, A. F. T., N. A. Smith, E. P. Xing, P. M. Q. Aguiar, and M. A. T. Figueiredo (2010). Turbo parsers: dependency parsing by approximate variational inference. In *Proceedings of EMNLP*. [19]

McAllester, D. A., T. Hazan, and J. Keshet (2010). Direct loss minimization for structured prediction. In *NIPS*. [49]

McCallum, A. and W. Li (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of CoNLL*. [23]

McClosky, D., E. Charniak, and M. Johnson (2006). Reranking and self-training for parser adaptation. In *Proceedings of ACL*. [150]

McDonald, R. (2006). *Discriminative Training and Spanning Tree Algorithms for Dependency Parsing*. Ph. D. thesis, University of Pennsylvania, Philadelphia, PA, USA. [17, 19]

McDonald, R., K. Crammer, and F. Pereira (2005). Online large-margin training of dependency parsers. In *Proceedings of ACL*. [18, 19, 35, 143, 146, 154]

McDonald, R., K. Hall, and G. Mann (2010). Distributed training strategies for the structured perceptron. In *Proceedings of HLT*. [54]

McDonald, R., K. Hannan, T. Neylon, M. Wells, and J. Reynar (2007). Structured models for fine-to-coarse sentiment analysis. In *Proceedings of ACL*. [78, 83, 84, 85, 86, 94, 96]

McDonald, R. and J. Nivre (2007). Characterizing the errors of data-driven dependency parsing models. In *Proceedings of EMNLP-CoNLL*. [18, 20]

McDonald, R. and F. Pereira (2006). Online learning of approximate dependency parsing algorithms. In *Proceedings of EACL*. [18, 19, 36]

McDonald, R., S. Petrov, and K. Hall (2011). Multi-source transfer of delexicalized dependency parsers. In *Proceedings of EMNLP*. [108, 113, 119, 120, 121, 132, 137, 138, 142]

Mei, Q., X. Ling, M. Wondra, H. Su, and C. Zhai (2007). Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proceedings of WWW*. [102]

Melamed, I. D. (2003). Multitext grammars and synchronous parsers. In *Proceedings of HLT-NAACL*. [109]

Mel'čuk, I. A. (1988). *Dependency syntax: Theory and practice*. SUNY. [17]

Miller, S., J. Guinness, and A. Zamanian (2004). Name tagging with word clusters and discriminative training. In *Proceedings of HLT-NAACL*. [64, 126, 132]

Minka, T. (2005). Discriminative models, not discriminative training. Technical Report MSR-TR-2005-144, Microsoft Research. [39]

Mitchell, T. (1997). *Machine Learning*. McGraw Hill. [29]

Mittrapiyanuruk, P. and V. Sornlertlamvanich (2000). The automatic Thai sentence extraction. In *Proceedings of SNLP*. [12]

Mnih, A. and G. Hinton (2007). Three new graphical models for statistical language modelling. In *Proceedings of ICML*. [127]

Mohammad, S., C. Dunne, and B. Dorr (2009). Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *Proceedings of EMNLP*. [26, 81]

Mohri, M. (2002). Semiring frameworks and algorithms for shortest-distance problems. *Automata, Languages and Combinatorics 7*(3), 321–350. [44]

Moon, T. and J. Baldridge (2007). Part-of-speech tagging for middle English through alignment and projection of parallel diachronic texts. In *Proceedings of EMNLP-CONLL*. [118]

Mulder, M., A. Nijholt, M. den Uyl, and P. Terpstra (2004). A lexical grammatical implementation of affect. In *Proceedings of TSD*. [24, 26]

Nadeau, D. and S. Sekine (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes 30*(1), 3–26. [21]

Nakagawa, T., K. Inui, and S. Kurohashi (2010). Dependency tree-based sentiment classification using CRFs with hidden variables. In *Proceedings of NAACL*. [15, 85, 87, 94, 99, 185]

Naseem, T., R. Barzilay, and A. Globerson (2012). Selective sharing for multilingual dependency parsing. In *Proceedings of ACL*. [5, 108, 113, 121, 142, 143, 144, 145, 146, 147, 148, 150, 154, 155, 183, 186]

Naseem, T., H. Chen, R. Barzilay, and M. Johnson (2010). Using universal linguistic knowledge to guide grammar induction. In *Proceedings of EMNLP*. [ix, 108, 111, 119, 123, 138, 145]

Naseem, T., B. Snyder, J. Eisenstein, and R. Barzilay (2009). Multilingual part-of-speech tagging: Two unsupervised approaches. *Journal of AI Research 36*(1), 341–385. [110, 159]

Neal, R. M. and G. E. Hinton (1999). A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan (Ed.), *Learning in graphical models*, pp. 355–368. Cambridge, MA, USA: MIT Press. [74]

Neuhaus, P. and N. Bröker (1997). The complexity of recognition of linguistically adequate dependency grammars. In *Proceedings of ACL*. [19]

Ng, A. Y. and M. I. Jordan (2001). On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Proceedings of NIPS*, Volume 13. [39]

Nigam, K., A. McCallum, S. Thrun, and T. Mitchell (1999). Text classification from labeled and unlabeled documents using EM. *Machine Learning 39*, 103–134. [65]

Nilsson, J., J. Nivre, and J. Hall (2007). Generalizing tree transformations for inductive dependency parsing. In *ACL*. [122]

Niu, F., B. Recht, C. Ré, and S. J. Wright (2011). Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. In *Proceedings of NIPS*. [54]

Nivre, J. (2003). An efficient algorithm for projective dependency parsing. In *Proceedings of IWPT*. [19, 20]

Nivre, J. (2006). *Inductive Dependency Parsing*. Number 34 in Text, Speech and Language Technology. New York, NY, USA: Springer-Verlag. [15, 16, 17, 121]

Nivre, J. (2008). Algorithms for deterministic incremental dependency parsing. *Computational Linguistics 34*(4), 513–553. [129]

Nivre, J., J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret (2007). The CoNLL 2007 shared task on dependency parsing. In *Proceedings of EMNLP-CoNLL*. [107, 110, 128, 166]

Nivre, J. and J. Nilsson (2005). Pseudo-projective dependency parsing. In *Proceedings of ACL*. [19]

Ohta, T., Y. Tateisi, and J.-D. Kim (2002). The genia corpus: an annotated research abstract corpus in molecular biology domain. In *Proceedings of HLT*. [22]

Osgood, C. E., G. J. Suci, and P. H. Tannenbaum (1967). *The Measurement of Meaning*. University of Illinois Press. [24]

Øvrelid, L. (2009). Cross-lingual porting of distributional semantic classification. In *Proceedings of NODALIDA*. [113, 119]

Padó, S. and M. Lapata (2006). Optimal constituent alignment with edge covers for semantic projection. In *Proceedings of ACL*. [113]

Pang, B. and L. Lee (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL*. [25, 26]

Pang, B. and L. Lee (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*. [25]

Pang, B. and L. Lee (2008). *Opinion mining and sentiment analysis*. Now Publishers. [23, 24, 78]

Pang, B., L. Lee, and S. Vaithyanathan (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of EMNLP*. [25, 26]

Peng, F., F. Feng, and A. McCallum (2004). Chinese segmentation and new word detection using conditional random fields. In *Proceedings of COLING*. [12]

Petrov, S., D. Das, and R. McDonald (2012). A universal part-of-speech tagset. In *Proceedings of LREC*. [13, 14, 34, 119, 123, 129, 137, 145, 167]

Pletscher, P. and P. Kohli (2012). Learning low-order models for enforcing high-order statistics. In *Proceedings of AISTATS*. [49]

Pletscher, P., C. S. Ong, and J. M. Buhmann (2010). Entropy and margin maximization for structured output learning. In *Proceedings of ECML-PKDD*. [52, 72]

Polyak, B. T. and A. B. Juditsky (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization 30*(4), 838–855. [57]

Popescu, A.-M. and O. Etzioni (2005). Extracting product features and opinions from reviews. In *Proceedings of EMNLP*. [25, 26]

Punyakanok, V., D. Roth, W.-t. Yih, and D. Zimak (2004). Semantic role labeling via integer linear programming inference. In *Proceedings of COLING*. [68]

Punyakanok, V., D. Roth, W.-t. Yih, and D. Zimak (2005). Learning and inference over constrained output. In *Proceedings of IJCAI*. [43, 68]

Qu, L., R. Gemulla, and G. Weikum (2012). A weakly supervised model for sentence-level semantic orientation analysis with multiple experts. In *Proceedings of EMNLP-CoNLL*. [102, 181]

Quattoni, A., S. Wang, L.-P. Morency, M. Collins, and T. Darrell (2007). Hidden conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*(10), 1848–1852. [72]

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning 1*(1), 81–106. [23]

Quirk, R., S. Greenbaum, G. Leech, and J. Svartvik (1985). *A Comprehensive Grammar of the English Language*. London, England: Longman. [23]

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of IEEE 77*(2), 257–286. [41]

Rakhlin, A., O. Shamir, and K. Sridharan (2012). Making stochastic gradient descent optimal for strongly convex problems. In *Proceedings of ICML*. [58]

Ramshaw, L. A. and M. P. Marcus (1995). Text chunking using transformation-based learning. In *Proceedings of VLC*. [22]

Rao, D. and D. Ravichandran (2009). Semi-supervised polarity lexicon induction. In *Proceedings of EACL*. [26, 103]

Ratinov, L. and D. Roth (2009). Design challenges and misconceptions in named entity recognition. In *Proceedings of CoNLL*. [22]

Ratinov, L., D. Roth, D. Downey, and M. Anderson (2011). Local and global algorithms for disambiguation to wikipedia. In *Proceedings of ACL-HLT*. [21]

Ratliff, N. (2009, May). *Learning to Search: Structured Prediction Techniques for Imitation Learning*. Ph. D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. [30]

Ratliff, N. D., A. J. Bagnell, and M. A. Zinkevich (2007). (Online) subgradient methods for structured prediction. In *Proceedings of AIStats*. [57]

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In *Proceedings of EMNLP*. [14, 33, 40]

Ravi, S. and K. Knight (2009). Minimized models for unsupervised part-of-speech tagging. In *Proceedings of ACL-IJCNLP*. [158, 159]

Ravi, S. and K. Knight (2011). Deciphering foreign language. In *Proceedings of ACL-HLT*. [186]

Razavi, M. (2012). Ensembles of diverse clustering-based discriminative dependency parsers. Master's thesis, School of Computing Science, Faculty of Applied Sciences, Simon Fraser University. [129, 185]

Reynar, J. C. and A. Ratnaparkhi (1997). A maximum entropy approach to identifying sentence boundaries. In *Proceedings of ANLC*. [12]

Ribarov, K. (2004). *Automatic building of a dependency tree.* Ph. D. thesis, Charles University. [18]

Riedel, S. and J. Clarke (2006). Incremental integer linear programming for non-projective dependency parsing. In *Proceedings of EMNLP.* [18, 43]

Riedel, S., D. Smith, and A. McCallum (2012). Parse, price and cut: delayed column and row generation for graph based parsers. In *Proceedings of EMNLP-CoNLL.* [19]

Riezler, S., T. H. King, R. M. Kaplan, R. Crouch, J. T. Maxwell, III, and M. Johnson (2002). Parsing the Wall Street Journal using a lexical-functional grammar and discriminative estimation techniques. In *Proceedings of ACL.* [150, 151]

Riloff, E. and J. Wiebe (2003). Learning extraction patterns for subjective expressions. In *Proceedings of EMNLP.* [26]

Riloff, E., J. Wiebe, and T. Wilson (2003). Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of CONLL.* [25]

Robbins, H. and S. Monro (1951). A stochastic approximation method. *Annals of Mathematical Statistics 22*(3), 400–407. [56, 57]

Robins, R. H. (1967). *A Short History of Linguistics.* Longman. [12]

Rosenblatt, F. (1958). The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review 65*, 386–408. [50]

Roth, D. and W. tau Yih (2004). A linear programming formulation for global inference in natural language tasks. In *CoNLL.* [68]

Ruppert, D. (1988). Efficient estimations from a slowly convergent Robbins-Monro process. Technical Report 781, Cornell University Operations Research and Industrial Engineering. [57]

Rush, A., R. Reichart, M. Collins, and A. Globerson (2012). Improved parsing and POS tagging using inter-sentence consistency constraints. In *Proceedings of EMNLP-CoNLL.* [33]

Rush, A. M. and S. Petrov (2012). Vine pruning for efficient multi-pass dependency parsing. In *Proceedings of NAACL-HLT.* [18, 34]

Saers, M. (2011). *Translation as Linear Transduction : Models and Algorithms for Efficient Learning in Statistical Machine Translation.* Ph. D. thesis, Uppsala University, Department of Linguistics and Philology, Uppsala, Sweden. [109]

Sag, I., T. Wasow, and E. Bender (2003). *Syntactic Theory: A Formal Introduction, 2nd Edition.* CSLI Lecture Notes Series. CSLI Publ. [16]

Sagae, K. and A. Lavie (2006). Parser combination by reparsing. In *Proceedings of NAACL.* [152]

Sagae, K. and J. Tsujii (2007). Dependency parsing and domain adaptation with LR models and parser ensembles. In *Proceedings of EMNLP-CoNLL.* [152]

Sahlgren, M., J. Karlgren, and G. Eriksson (2007). SICS: Valence annotation based on seeds in word space. In *Proceedings of SemEval.* [24, 25, 26]

Salakhutdinov, R., S. T. Roweis, and Z. Ghahramani (2003). Optimization with EM and expectation-conjugate-gradient. In *Proceedings of ICML.* [74, 75]

Samdani, R., M.-W. Chang, and D. Roth (2012). Unified expectation maximization. In *Proceedings of NAACL.* [69, 72]

Saul, L. K. and S. T. Roweis (2003). Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research 4*, 119–155. [64]

Sauper, C., A. Haghighi, and R. Barzilay (2010). Incorporating content structure into text analysis applications. In *Proceedings of EMNLP*. [102]

Schaul, T., S. Zhang, and Y. LeCun (2012). No more pesky learning rates. *arXiv:1206.1106v2 [stat.ML]*. [57]

Schone, P. and D. Jurafsky (2001). Language-independent induction of part of speech class labels using only language universals. In *Proceedings of the IJCAI Workshop on Text Learning: Beyond Supervision*. [111]

Schraudolph, N. N. (1999). Local gain adaptation in stochastic gradient descent. In *Proceedings of ICANN*. [57]

Schrijver, A. (1998). *Theory of Linear and Integer Programming*. Wiley Series in Discrete Mathematics & Optimization. New York, NY, USA: John Wiley & Sons. [18]

Schütze, H. (1993). Part-of-speech induction from scratch. In *Proceedings of ACL*. [66, 127, 159]

Schütze, H. (1995). Distributional part-of-speech tagging. In *Proceedings of EACL*. [127, 159]

Schwartz, R., O. Abend, R. Reichart, and A. Rappoport (2011). Neutralizing linguistically problematic annotations in unsupervised dependency parsing evaluation. In *Proceedings of ACL-HLT*. [123]

Seddah, D., S. Koebler, and R. Tsarfaty (Eds.) (2010). *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*. Los Angeles, CA, USA: Association for Computational Linguistics. [107]

Seddah, D., R. Tsarfaty, and J. Foster (Eds.) (2011). *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*. Dublin, Ireland: Association for Computational Linguistics. [107]

Sekine, S. and C. Nobata (2004). Definition, dictionaries and tagger for extended named entity hierarchy. In *Proceedings of LREC*. [22]

Shalev-Shwartz, S. (2012). *Online Learning and Online Convex Optimization*. Now Publishers. [54]

Shalev-Shwartz, S., Y. Singer, N. Srebro, and A. Cotter (2011). Pegasos: primal estimated sub-gradient solver for SVM. *Mathematical Programming 127*(1), 3–30. [56]

Shannon, C. (1948). A mathematical theory of communication. *Bell System Technical Journal 27*(3), 379–423, 623–656. [3]

Shannon, C. (1951). Prediction and entropy of printed English. *Bell System Technical Journal 30*, 50–64. [3]

Skjærholt, A. and L. Øvrelid (2011). Impact of treebank characteristics on cross-lingual parser adaptation. In *Proceedings of TLT*. [119]

Smith, D. A. and J. Eisner (2006). Quasi-synchronous grammars: alignment by soft projection of syntactic dependencies. In *Proceedings of the Workshop on Statistical Machine Translation*. [118]

Smith, D. A. and J. Eisner (2008). Dependency parsing by belief propagation. In *Proceedings of EMNLP*. [18, 36]

Smith, D. A. and J. Eisner (2009). Parser adaptation and projection with quasi-synchronous grammar features. In *Proceedings of EMNLP*. [113, 118, 121, 187]

Smith, D. A. and N. A. Smith (2004). Bilingual parsing with factored estimation: Using English to parse Korean. In *Proceedings of EMNLP*. [109, 110]

Smith, N. A. (2011). *Linguistic Structure Prediction*. Morgan & Claypool Publishers. [11, 29, 33]

Smith, N. A. and J. Eisner (2005a). Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of ACL*. [69, 158, 159]

Smith, N. A. and J. Eisner (2005b). Guiding unsupervised grammar induction using contrastive estimation. In *Proceedings of IJCAI Workshop on Grammatical Inference Applications*. [69]

Snyder, B. and R. Barzilay (2007). Multiple aspect ranking using the Good Grief algorithm. In *Proceedings of NAACL-HLT*. [26]

Snyder, B. and R. Barzilay (2008). Unsupervised multilingual learning for morphological segmentation. In *Proceedings of ACL-HLT*. [111]

Snyder, B. and R. Barzilay (2010). Climbing the tower of Babel: Unsupervised multilingual learning. In *Proceedings of ICML*. [120, 141]

Snyder, B., T. Naseem, and R. Barzilay (2009). Unsupervised multilingual grammar induction. In *Proceedings of ACL-IJCNLP*. [109, 110]

Snyder, B., T. Naseem, J. Eisenstein, and R. Barzilay (2008). Unsupervised multilingual learning for POS tagging. In *Proceedings of EMNLP*. [110]

Socher, R., J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning (2011). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of EMNLP*. [185]

Søgaard, A. (2011). Data point selection for cross-language adaptation of dependency parsers. In *Proceedings of ACL*. [113, 119, 120, 121, 142, 186]

Søgaard, A. and J. Wulff (2012). An empirical study of non-lexical extensions to delexicalized transfer. In *Proceedings of COLING*. [113, 121, 142, 186]

Spencer, A. and A. M. Zwicky (2001). *The Handbook of Morphology*. Blackwell Handbooks in Linguistics. Oxford, UK: Wiley. [12]

Spitkovsky, V. I., H. Alshawi, A. X. Chang, and D. Jurafsky (2011). Unsupervised dependency parsing without gold part-of-speech tags. In *Proceedings of EMNLP*. [108]

Spitkovsky, V. I., H. Alshawi, and D. Jurafsky (2012). Three dependency-and-boundary models for grammar induction. In *Proceedings of EMNLP-CoNLL*. [108]

Spreyer, K. (2010). Notes on the evaluation of dependency parsers obtained through cross-lingual projection. In *Proceedings of COLING*. [122]

Spreyer, K. (2011). *Does it have to be trees? Data-driven dependency parsing with incomplete and noisy training data*. Ph. D. thesis, University of Potsdam, Potsdam, Germany. [115, 116, 117]

Spreyer, K. and J. Kuhn (2009). Data-driven dependency parsing of new languages using incomplete and noisy training data. In *Proceedings of CONLL*. [113]

Sproat, R., W. Gale, C. Shih, and N. Chang (1996). A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics 22*(3), 377–404. [12]

Steinberger, R., B. Pouliquen, M. Kabadjov, J. Belyaeva, and E. van der Goot (2011). JRC-NAMES: A freely available, highly multilingual named entity resource. In

Proceedings of RANLP. [22, 23]

Stone, P. J., D. C. Dunphy, M. S. Smith, and D. M. Ogilvie (1966). *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press. [23, 24]

Stymne, S. (2012). *Text Harmonization Strategies for Phrase-Based Statistical Machine Translation*. Ph. D. thesis, Linköping University, NLPLAB - Natural Language Processing Laboratory, The Institute of Technology, Linköping, Sweden. [115]

Subasic, P. and A. Huettner (2001). Affect analysis of text using fuzzy semantic typing. *IEEE Transaction on Fuzzy Systems 9*(4), 483–496. [24]

Suzuki, J., A. Fujino, and H. Isozaki (2007). Semi-supervised structured output learning based on a hybrid generative and discriminative approach. In *Porceedings of EMNLP*. [102]

Täckström, O. (2012). Nudging the envelope of direct transfer methods for multilingual named entity recognition. In *Proceedings of the NAACL-HLT Workshop on Inducing Linguistic Structure (WILS)*. [8, 150]

Täckström, O., D. Das, S. Petrov, R. McDonald, and J. Nivre (2013). Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics 1*, 1–12. [8]

Täckström, O. and R. McDonald (2011a). Discovering fine-grained sentiment with latent variable structured prediction models. In *Proceedings of ECIR*. [8, 102]

Täckström, O. and R. McDonald (2011b). Discovering fine-grained sentiment with latent variable structured prediction models. Technical Report T2011:02, Swedish Institute of Computer Science (SICS), Kista, Sweden. [8]

Täckström, O. and R. McDonald (2011c). Semi-supervised latent variable models for sentence-level sentiment analysis. In *Proceedings of ACL-HLT*. [8]

Täckström, O., R. McDonald, and J. Nivre (2013). Target language adaptation of discriminative transfer parsers. In *Proceedings of NAACL-HLT*. [8]

Täckström, O., R. McDonald, and J. Uszkoreit (2012). Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of NAACL-HLT*. [8]

Tapanainen, P. and T. Järvinen (1997). A non-projective dependency parser. In *Proceedings of ANLP*. [18]

Tarlow, D. and R. S. Zemel (2012). Structured output learning with high order loss functions. In *Proceedings of AISTATS*. [49]

Taskar, B. (2004). *Learning Structured Prediction Models: A Large Margin Approach*. Ph. D. thesis, Stanford University, Stanford, CA, USA. [29]

Taskar, B., C. Guestrin, and D. Koller (2003). Max-margin Markov networks. In *Proceedings of NIPS*. [49, 51, 52]

Tesniére, L. (1959). *Éléments de syntaxe structurale*. Editions Klincksieck. [16]

Thomas, M., B. Pang, and L. Lee (2006). Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *Proceedings of EMNLP*. [24]

Tibshirani, R. (1994). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B 58*, 267–288. [53]

Tiedemann, J. (2011). *Bitext Alignment*. Synthesis Lectures on Human Language Technologies Series. Morgan & Claypool Pub. [115]

Tikhonov, A. and V. Arsenin (1977). *Solutions of ill posed problems*. Washington, DC, USA: Vh Winston & Sons. [53]

Tim Berners-Lee, J. H. and O. Lassila (2001). The semantic web. [21]

Titov, I. and A. Klementiev (2012). Crosslingual induction of semantic roles. In *Proceedings of ACL*. [111]

Titov, I. and R. McDonald (2008). Modeling online reviews with multi-grain topic models. In *Proceedings of WWW*. [26, 102]

Tjong Kim Sang, E. F. (2002). Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL*. [22, 34, 48, 107, 109, 129]

Tjong Kim Sang, E. F. and F. De Meulder (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL*. [22, 34, 48, 107, 110, 129]

Toutanova, K., A. Haghighi, and C. Manning (2005). Joint learning improves semantic role labeling. In *Proceedings of ACL*. [68]

Tratz, S. and E. Hovy (2011). A fast, effective, non-projective, semantically-enriched parser. In *Proceedings of EMNLP*. [126]

Tsarfaty, R., J. Nivre, and E. Andersson (2011). Evaluating dependency parsing: Robust and heuristics-free cross-annotation evaluation. In *Proceedings of EMNLP*. [123]

Tsochantaridis, I., T. Joachims, T. Hofmann, and Y. Altun (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research 6*, 1453–1484. [49, 51, 52]

Tsuruoka, Y., J. Tsujii, and S. Ananiadou (2009). Stochastic gradient descent training for L1-regularized log-linear models with cumulative penalty. In *Proceedings of ACL-AFNLP*. [56]

Tu, K. and V. Honavar (2012). Unambiguity regularization for unsupervised learning of probabilistic grammars. In *Proceedings of EMNLP-CoNLL*. [72, 108]

Turian, J., L.-A. Ratinov, and Y. Bengio (2010). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of ACL*. [64, 126, 127, 129, 130, 132, 185]

Turney, P. (2002). Thumbs up or thumbs down? Sentiment orientation applied to unsupervised classification of reviews. In *Proceedings of ACL*. [25, 26, 103]

Turney, P. D. and M. L. Littman (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems 21*(4), 315–346. [24]

Tutte, W. T. (2001). *Graph Theory*. Cambridge University Press. [17]

UN (2006). ODS UN parallel corpus. [167]

Uszkoreit, J. and T. Brants (2008). Distributed word clustering for large scale class-based language modeling in machine translation. In *Proceedings of ACL-HLT*. [126, 127, 129]

Uszkoreit, J., J. Ponte, A. Popat, and M. Dubiner (2010). Large scale parallel document mining for machine translation. In *Proceedings of COLING*. [167]

van Rijsbergen, C. J. (1979). *Information Retrieval* (2nd ed.). Newton, MA, USA: Butterworth-Heinemann. [48]

Van Valin, R. D. (2001). *An Introduction to Syntax.* Cambridge, UK: Cambridge University Press. [12, 16]

Vapnik, V. (1998). *Statistical Learning Theory.* New York, NY, USA: John Wiley & Sons. [37, 46, 47, 51]

Vapnik, V. N. and A. J. Chervonenkis (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications 16*(2), 264–280. [46]

Velikovich, L., S. Blair-Goldensohn, K. Hannan, and R. McDonald (2010). The viability of web-derived polarity lexicons. In *Proceedings of NAACL.* [26, 66, 81, 82, 103]

Vishwanathan, S. V. N., N. N. Schraudolph, M. W. Schmidt, and K. P. Murphy (2006). Accelerated training of conditional random fields with stochastic gradient methods. In *Proceedings of ICML.* [57]

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory 13*(2), 260–269. [44]

Wainwright, M. J. and M. I. Jordan (2008). *Graphical Models, Exponential Families, and Variational Inference.* Now Publishers. [18, 40]

Wakao, T., R. Gaizauskas, and Y. Wilks (1996). Evaluation of an algorithm for the recognition and classification of proper names. In *Proceedings COLING.* [22]

Weinberger, K., A. Dasgupta, J. Langford, A. Smola, and J. Attenberg (2009). Feature hashing for large scale multitask learning. In *Proceedings of ICML.* [60]

Weiss, D. and B. Taskar (2010). Structured prediction cascades. In *Proceedings of AISTATS.* [18, 34]

Weston, J., F. Ratle, and R. Collobert (2008). Deep learning via semi-supervised embedding. In *Proceedings of ICML.* [65]

Whitelaw, C., A. Kehlenbeck, N. Petrovic, and L. Ungar (2008). Web-scale named entity recognition. In *Proceedings of CIKM.* [21]

Wiebe, J. (2000). Learning subjective adjectives from corpora. In *Proceedings of AAI.* [26]

Wiebe, J., T. Wilson, R. F. Bruce, M. Bell, and M. Martin (2004). Learning subjective language. *Computational Linguistics 30*(3), 277–308. [23, 25, 26]

Wiebe, J., T. Wilson, and C. Cardie (2005). Annotating expressions of opinions and emotions in language. In *Proceedings of LREC.* [26, 27, 77, 102]

Wilks, Y. and C. Brewster (2009). Natural language processing as a foundation of the semantic web. *Foundations and Trends in Web Science 1*(3-4), 199–327. [21]

Williams, P. M. (1995). Bayesian regularization and pruning using a Laplace prior. *Neural Computation 7*(1), 117–143. [53]

Wilson, T., J. Wiebe, and P. Hoffmann (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of EMNLP.* [26, 80, 81, 82]

Wu, D. (1997). Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics 23*(3), 377–403. [109]

Xi, C. and R. Hwa (2005). A backoff model for bootstrapping resources for non-English languages. In *Proceedings of HLT-EMNLP.* [157, 158]

Xue, N., F. Xia, F.-d. Chiou, and M. Palmer (2005). The Penn Chinese treebank: Phrase structure annotation of a large corpus. *Natural Language*

Engineering 11(02), 207–238. [128]

Yamada, H. and Y. Matsumoto (2003). Statistical dependency analysis with support vector machines. In *Proceedings of IWPT*. [17, 19, 20, 30]

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of ACL*. [65, 66]

Yarowsky, D. and G. Ngai (2001). Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of NAACL*. [111, 113, 114, 116, 118, 157, 158, 187]

Yarowsky, D., G. Ngai, and R. Wicentowski (2001). Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of HLT*. [108, 111, 113, 114, 117]

Yedidia, J. S., W. T. Freeman, and Y. Weiss (2003). Understanding belief propagation and its generalizations. In G. Lakemeyer and B. Nebel (Eds.), *Exploring artificial intelligence in the new millennium*, pp. 239–269. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. [44]

Yessenalina, A., Y. Yue, and C. Cardie (2010). Multi-level structured models for document-level sentiment classification. In *Proceedings of EMNLP*. [85, 90, 94, 99, 102, 180]

Yih, W., P. Chang, and W. Kim (2004). Mining online deal forums for hot deals. In *Proceedings of WebIntelligence*. [25]

Yu, C.-N. and T. Joachims (2009). Learning structural SVMs with latent variables. In *Proceedings of ICML*. [72]

Zeman, D., D. Mareček, M. Popel, L. Ramasamy, J. Štěpánek, Z. Žabokrtský, and J. Hajič (2012). HamleDT: To parse or not to parse? In *Proceedings of LREC*. [122, 123]

Zeman, D. and P. Resnik (2008). Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP Workshop: NLP for Less Privileged Languages*. [113, 119, 120, 123, 132, 150]

Zhang, H. and R. McDonald (2012). Generalized higher-order dependency parsing with cube pruning. In *Proceedings of EMNLP-CoNLL*. [18, 19, 36]

Zhang, T. (2004). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of ICML*. [54]

Zhang, Y. and S. Clark (2008). A tale of two parsers: investigating and combining graph-based and transition-based dependency parsing using beam-search. In *Proceedings of EMNLP*. [20, 129]

Zhang, Y. and J. Nivre (2011). Transition-based dependency parsing with rich non-local features. In *Proceedings of ACL-HLT*. [20, 130]

Zhang, Y., R. Reichart, R. Barzilay, and A. Globerson (2012). Learning to map into a universal POS tagset. In *Proceedings of EMNLP-CoNLL*. [111, 123]

Zhou, D., O. Bousquet, T. Lal, J. Weston, and B. Schölkopf (2003). Learning with local and global consistency. In *Proceedings of NIPS*. [65]

Zhu, X., Z. Ghahramani, and J. Lafferty (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of ICML*. [65, 118]

Zhuang, L., F. Jing, and X.-Y. Zhu (2006). Movie review mining and summarization. In *Proceedings of CIKM*. [26]

Zou, H. and T. Hastie (2005). Regularization and variable selection via the Elastic Net. *Journal of the Royal Statistical Society, Series B 67*, 301–320. [53]

Appendix A. Language Codes

The table below lists the two-letter ISO 639-1 code for each language studied in the dissertation.[1]

Code	Language
ar	Arabic
bg	Bulgarian
ca	Catalan
cs	Czech
da	Danish
de	German
el	Greek
en	English
es	Spanish
eu	Basque
fr	French
hu	Hungarian
it	Italian
ja	Japanese
ko	Korean
nl	Dutch
pt	Portuguese
ru	Russian
sl	Slovene
sv	Swedish
tr	Turkish
zh	Chinese

[1] Obtained from http://www.loc.gov/standards/iso639-2 — February 14, 2013.

ACTA UNIVERSITATIS UPSALIENSIS
Studia Linguistica Upsaliensia
Department of Linguistics
Editors: Joakim Nivre and Åke Viberg

1. *Jörg Tiedemann*, Recycling translations. Extraction of lexical data from parallel corpora and their application in natural language processing. 2003.
2. *Agnes Edling*, Abstraction and authority in textbooks. The textual paths towards specialized language. 2006.
3. *Åsa af Geijerstam*, Att skriva i naturorienterande ämnen i skolan. 2006.
4. *Gustav Öquist*, Evaluating Readability on Mobile Devices. 2006.
5. *Jenny Wiksten Folkeryd*, Writing with an Attitude. Appraisal and student texts in the school subject of Swedish. 2006.
6. *Ingrid Björk*, Relativizing linguistic relativity. Investigating underlying assumptions about language in the neo-Whorfian literature. 2008.
7. *Joakim Nivre, Mats Dahllöf and Beáta Megyesi*, Resourceful Language Technology. Festschrift in Honor of Anna Sågvall Hein. 2008.
8. *Anju Saxena and Åke Viberg*, Multilingualism. Proceedings of the 23rd Scandinavian Conference of Linguistics. 2009.
9. *Markus Saers*, Translation as Linear Transduction. Models and Algorithms for Efficient Learning in Statistical Machine Translation. 2011.
10. *Ulrika Serrander*, Bilingual lexical processing in single word production. Swedish learners of Spanish and the effects of L2 immersion. 2011.
11. *Mattias Nilsson*, Computational Models of Eye Movements in Reading: A Data-Driven Approach to the Eye-Mind Link. 2012.
12. *Luying Wang*, Second Language Acquisition of Mandarin Aspect Markers by Native Swedish Adults. 2012.
13. *Farideh Okati*, The Vowel Systems of Five Iranian Balochi Dialects. 2012.
14. *Oscar Täckström*, Predicting Linguistic Structure with Incomplete and Cross-Lingual Supervision. 2013.

SICS Dissertation Series
SICS Swedish ICT

1. *Bogumil Hausman*, Pruning and Speculative Work in OR-Parallel PROLOG. 1990.
2. *Mats Carlsson*, Design and Implementation of an OR-Parallel Prolog Engine. 1990.
3. *Nabiel A. Elshiewy*, Robust Coordinated Reactive Computing in SANDRA. 1990.
4. *Dan Sahlin*, An Automatic Partial Evaluator for Full Prolog. 1991.
5. *Hans A. Hansson*, Time and Probability in Formal Design of Distributed Systems, 1991.
6. *Peter Sjödin*, From LOTOS Specifications to Distributed Implementations. 1991.
7. *Roland Karlsson*, A High Performance OR-Parallel Prolog System. 1992.
8. *Erik Hagersten*, Toward Scalable Cache Only Memory Architectures. 1992.
9. *Lars-Henrik Eriksson*, Finitary Partial Inductive Definitions and General Logic. 1993.
10. *Mats Björkman*, Architectures for High Performance Communication. 1993.
11. *Stephen Pink*, Measurement, Implementation, and Optimization of Internet Protocols. 1993.
12. *Martin Aronsson*, GCLA. The Design, Use, and Implementation of a Program Development System. 1993.
13. *Christer Samuelsson*, Fast Natural-Language Parsing Using Explanation-Based Learning. 1994.
14. *Sverker Jansson*, AKL: A Multiparadigm Programming Language. 1994.
15. *Fredrik Orava*, On the Formal Analysis of Telecommunication Protocols. 1994.
16. *Torbjörn Keisu*, Tree Constraints. 1994.
17. *Olof Hagsand*, Computer and Communication Support for Interactive Distributed Applications. 1995.
18. *Björn Carlsson*, Compiling and Executing Finite Domain Constraints. 1995.
19. *Per Kreuger*, Computational Issues in Calculi of Partial Inductive Definitions. 1995.
20. *Annika Waern*, Recognising Human Plans: Issues for Plan Recognition in Human-Computer Interaction, 1996.
21. *Björn Gambäck*, Processing Swedish Sentences: A Unification-Based Grammar and Some Applications. 1997.
22. *Klas Orsvärn*, Knowledge Modelling with Libraries of Task Decomposition Methods. 1996.
23. *Kia Höök*, A Glass Box Approach to Adaptive Hypermedia. 1996.
24. *Bengt Ahlgren*, Improving Computer Communication Performance by Reducing Memory Bandwidth Consumption. 1997.
25. *Johan Montelius*, Exploiting Fine-grain Parallelism in Concurrent Constraint Languages. 1997.
26. *Jussi Karlgren*, Stylistic experiments in information retrieval. 2000.
27. *Ashley Saulsbury*, Attacking Latency Bottlenecks in Distributed Shared Memory Systems. 1999.

28. *Kristian Simsarian*, Toward Human Robot Collaboration. 2000.
29. *Lars-åke Fredlund*, A Framework for Reasoning about Erlang Code. 2001.
30. *Thiemo Voigt*, Architectures for Service Differentiation in Overloaded Internet Servers. 2002.
31. *Fredrik Espinoza*, Individual Service Provisioning. 2003.
32. *Lars Rasmusson*, Network capacity sharing with QoS as a financial derivative pricing problem: algorithms and network design. 2002.
33. *Martin Svensson*, Defining, Designing and Evaluating Social Navigation. 2003.
34. *Joe Armstrong*, Making reliable distributed systems in the presence of software errors. 2003.
35. *Emmanuel Frécon*, DIVE on the Internet. 2004.
36. *Rickard Cöster*, Algorithms and Representations for Personalised Information Access. 2005.
37. *Per Brand*, The Design Philosophy of Distributed Programming Systems: the Mozart Experience. 2005.
38. *Sameh El-Ansary*, Designs and Analyses in Structured Peer-to-Peer Systems. 2005.
39. *Erik Klintskog*, Generic Distribution Support for Programming Systems. 2005.
40. *Markus Bylund*, A Design Rationale for Pervasive Computing: User Experience, Contextual Change, and Technical Requirements. 2005.
41. *Åsa Rudström*, Co-Construction of hybrid spaces. 2005.
42. *Babak Sadighi Firozabadi*, Decentralised Privilege Management for Access Control. 2005.
43. *Marie Sjölinder*, Age-related Cognitive Decline and Navigation in Electronic Environments. 2006.
44. *Magnus Sahlgren*, The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-dimensional Vector Spaces. 2006.
45. *Ali Ghodsi*, Distributed k-ary System: Algorithms for Distributed Hash Tables. 2006.
46. *Stina Nylander*, Design and Implementation of Multi-Device Services. 2007.
47. *Adam Dunkels*, Programming Memory-Constrained Networked Embedded Systems. 2007.
48. *Jarmo Laaksolahti*, Plot, Spectacle, and Experience: Contributions to the Design and Evaluation of Interactive Storytelling. 2008.
49. *Daniel Gillblad*, On Practical Machine Learning and Data Analysis. 2008.
50. *Fredrik Olsson*, Bootstrapping Named Entity Annotation by Means of Active Machine Learning: a Method for Creating Corpora. 2008.
51. *Ian Marsh*, Quality Aspects of Internet Telephony. 2009.
52. *Markus Bohlin*, A Study of Combinatorial Optimization Problems in Industrial Computer Systems. 2009.
53. *Petra Sundström*, Designing Affective Loop Experiences. 2010.
54. *Anders Gunnar*, Aspects of Proactive Traffic Engineering in IP Networks. 2011.
55. *Preben Hansen*, Task-based Information Seeking and Retrieval in the Patent Domain: Process and Relationships. 2011.

56. *Fredrik Österlind*, Improving Low-Power Wireless Protocols with Timing-Accurate Simulation. 2011.
57. *Ahmad Al-Shishtawy*, Self-Management for Large-Scale Distributed Systems. 2012.
58. *Henrik Abrahamsson*, Network overload avoidance by traffic engineering and content caching. 2012.
59. *Mattias Rost*, Mobility is the Message: Experiment with Mobile Media Sharing. 2013.
60. *Amir H. Payberah*, Live Streaming in P2P and Hybrid P2P-Cloud Environments for the open Internet. 2013.
61. *Oscar Täckström*, Predicting Linguistic Structure with Incomplete and Cross-Lingual Supervision. 2013.