Linguistically Motivated Models for Lightly-Supervised Dependency Parsing

by

Tahira Naseem

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2014

Signature redacted

Author .. ..
Department of Electrical Engineering and Computer Science
February 14, 2014

Signature redacted

Certified by ....
Regina Barzilay
Professor, Electrical Engineering and Computer Science
Thesis Supervisor

Signature redacted

Accepted by..... .....
Leslie A. Kolodziejski
Professor and Chairman, Department Committee on Graduate Students

Linguistically Motivated Models for Lightly-Supervised Dependency Parsing

by

Tahira Naseem

Submitted to the Department of Electrical Engineering and Computer Science
on February 14, 2014, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Today, the top performing parsing algorithms rely on the availability of annotated data for learning the syntactic structure of a language. Unfortunately, syntactically annotated texts are available only for a handful of languages. The research presented in this thesis aims at developing parsing models that can effectively perform in a lightly-supervised training regime. In particular we focus on formulating linguistically aware models of dependency parsing that can exploit readily available sources of linguistic knowledge such as language universals and typological features. This type of linguistic knowledge can be used to motivate model design and/or to guide inference procedure.

We propose three alternative approaches for incorporating linguistic information into a lightly-supervised training setup:

First, we show that linguistic information can be used in the form of rules on top of standard unsupervised parsing models to guide inference procedure. This method consistently outperforms existing monolingual and multilingual unsupervised parsers when tested on a set of 6 Indo-European languages.

Next, we show that a linguistically aware model design greatly facilitates crosslingual parser transfer by leveraging syntactic connections between languages. Our transfer approach outperforms the state-of-the-art multilingual transfer parser across a set of 19 languages, achieving an average gain of 5.9%. The gains are even more pronounced – 14.4% – on non-Indo-European languages where existing transfer methods fail to perform.

Finally, we propose a corpus-level Bayesian framework that allows multiple views of data

in a single model. We use this framework to combine a dependency model with constituency view and universal rules, achieving a performance gain of 1.9% compared to the top-performing unsupervised parsing model.

Thesis Supervisor: Regina Barzilay
Title: Professor, Electrical Engineering and Computer Science

Acknowledgments

First of all I would like to thank my advisor Regina Barzilay for shaping me into a researcher. When I came here 6 years ago, I knew about research almost as much I as I now know about baseball[1]. Regina has a way of bringing out the best in her students. Her high standards and her insight into innovative and impactful directions of research has been the driving force behind my work. I am thankful to her for her guidance and support throughout the span of my PhD and especially for believing in me when I was not all that willing to believe in myself – I could not have asked for a better advisor.

I am also thankful to my wonderful thesis committee and my collaborators, Tommi Jaakkola, Ryan McDonald, Amir Globerson, Harr Chen and Mark Johnson. Without their input and guidance, this work would not have been possible.

I would also like to thank all my colleagues. First of all I would thank Jacob Eisenstein and Benjamin Snyder for partially filling in the huge void of knowledge that I always carry around with me. I would also like to thank Branavan and Yoonk Keok who somehow manage to remind me that researchers are also human beings. I am thankful to all my lab mates, Zach, Karthik, Nate, Yonatan, Tao, Yuan, Christy and Andreea for listening to my stories and patiently waiting for any hint of logic till the very end.

Finally, I am thankful to my wonderful and very interesting family for their love and support, especially to my mom who always has a hidden hand in everything I ever accomplish. I am also thankful to my brand new husband for the great person he will turn out to be during the rest of my life. I dedicate this thesis to my amazing family.

[1] I know that it is a game and it bears some resemblance with cricket.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Computer understanding of natural language can have enormous impact on the way we communicate today. It can lead towards automation of numerous tasks involving human-human and human-computer communication, ranging from question-answering to translation and text summarization. Most of the research in this area relies on some form of syntactic analysis of text. Consequently, automatic syntactic parsing has evolved into a major sub-task in the field of Natural Language Processing (NLP).

Today, the top performing parsing algorithms rely on the availability of annotated data for learning the syntactic structure of a language. Unfortunately, syntactically annotated texts are available only for a handful of languages. Moreover, the development of such resources for a new language can be both expensive and time consuming because of the level of linguistic expertise needed to produce syntactic annotations. However, there is a vast body of knowledge in the form of linguistic theories which remains mostly untapped in the context of grammar induction. Existing techniques that aim to benefit resource-lean languages either assume a fully unsupervised setup [47, 44, 20] or transfer syntactic information from annotated resources in other languages [73, 19, 58]. There is surprisingly little existing

work that makes use of available linguistic sources of knowledge. The work presented in this thesis exploits linguistic knowledge about syntax and its crosslingual variations when learning grammar. The aim is to develop parsing methods that are applicable across a wide range of human languages with minimal supervision.

History of linguistics dates back to 4th century BC starting from the Sanskrit grammar of Panini. Since then, a huge amount of knowledge has accumulated in the form of linguistic theories and grammar formalisms. This includes both language specific studies as well as general formalisms for representing languages. The popular forms of syntactic representation used in NLP today (e.g. Context Free Grammar, Dependency Grammar, Tree Adjoining Grammar) are based on established linguistic theories. When training a parsing model for any particular formalism, manually annotated treebanks (sentences annotated with syntactic structure) of the desired formalism for the language of interest are given to the parser.

Human annotators who produce treebanks for any particular formalism are provided with annotation guidelines. The guidelines are needed for two reasons: First, although the native speaker of a language can perfectly understand the meaning of the sentences, consciously marking syntactic relations between words is a difficult task even for humans. Secondly, in linguistic theories there are multiple views on how to represent certain very frequent constructs. For instance, in dependency parsing framework, there is still no consensus on the head of Noun phrases. Therefore, guidelines are considered necessary for correct and consistent annotation of data.

In contrast, unsupervised parsers are expected to learn the syntactic structure in the absence of any linguistic guidance. This setup would make sense if the aim was to discern patterns in language structure previously unidentified by linguists. However, that is not the case, standard practice is to evaluate these parsers against manual annotations. The parsers are

expected to automatically learn the structure that the annotators had in mind (or in their guidelines). Thus the primary goal of the research in grammar induction has been to save annotation effort. Our claim is that this goal can be achieved more effectively if, like the human annotators, the parsers are guided towards the desired annotations.

Readily available forms of linguistic knowledge include basic part-of-speech definitions, dominant syntactic properties such as right-branching vs. left-branching, and word order trends like Prepositions vs. Postpositions. Such high-level syntactic properties are well documented for most languages. Yet, unsupervised parsers try to re-learn those trends and fail to get even basic dependencies right. Supervised parsers, on the other hand, learn these patterns and much more from the detailed treebanks used for training. For instance, when trained on large treebanks, these parsers learn lexical selectional preferences. Encoding these preferences manually into a parser can be quite cumbersome if not impossible. This indicates that in order to bridge the gap between supervised and unsupervised methods, high-level linguistic knowledge must be combined with learning from data.

Adding linguistic knowledge to a data driven parsing models poses some challenges. The main challenge is that this knowledge is available in declarative forms without any notion of degree of applicability. For instance, it may tell us that a certain language is dominantly right-branching, but the degree of dominance remains unknown. Moreover, even when the information is almost always applicable, there is still room for ambiguity. For instance, if we know that a language is always Prepositional, in a given sentence there can still be multiple Prepositions preceding a Noun that can be potential heads. We explore different ways of balancing declarative knowledge with the patterns learned from data.

The work presented in this thesis shows the benefits of using linguistic knowledge in combination with learning from data. Since the first better-than-random parser in 2004 [47] (with an average accuracy of 33.6% [75]), unsupervised techniques have reached an accuracy of

48.6% [76]. However, our unsupervised parser, with access to a small set of linguistic rules, outperforms these models with an accuracy of 50.7%. Furthermore, our linguistically motivated transfer parser and semi-supervised parser achieve 59.5% and 67.2% respectively. The information provided to these parsers is minimal and readily available yet the gains are substantial when compared with fully unsupervised models.

We explore the use of different forms of linguistic information as well as different ways of incorporating this information into parsing models. First, we show that linguistic information can be used in the form of universal rules on top of standard unsupervised parsing models to guide inference procedure, yielding impressive gains in performance. Next, we show that a linguistically aware model design greatly facilitates crosslingual parser transfer by leveraging crosslingual syntactic connections. Finally, we propose a simple yet effective method that allows multiple linguistic views of data in a single model.

Before further elaborating on our approach, we first give a brief overview of dependency grammar formalism followed by a discussion of dependency parsing research in NLP. We then continue with a more detailed overview of the work presented in this thesis followed by an outline of the rest of the thesis.

1.1 Dependency Grammar Formalism

In modern linguistics, dependency based representation started with the work of Tesniere published in 1959 [48]. Since then a number of formalisms based on binary word dependencies have emerged. The fundamental difference between dependency base representation and more traditional constituency representation is the following: In dependency tree the number of tree nodes is equal to the number of words in the sentence, whereas in constituency tree, there maybe additional internal tree nodes specifying the syntactic labels of word spans. Thus the length of the sentence, in a way, limits the complexity of its dependency tree, which makes this formalism relatively simple and more suitable for computational analysis.

Figure 1-1 shows the Dependency based syntactic structure of a sentence. In dependency parse of a sentence, each word serves as a modifier of some other word in the sentence. In the figure, the arrow heads point to the modifier word. The word being modified is called the "head" or the "parent", and the modifier may be called "dependent" or "child". The basic idea is that a modifier word is present in the sentence because of its head word and is therefore dependent on its head word, moreover it modifies or adds to the meaning of its head. For instance, in the example sentence "red" is a modifier of "apples", its presence in

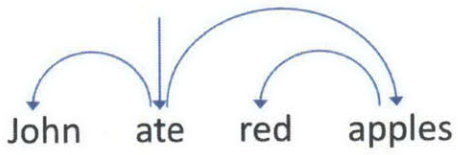


Figure 1-1: A Dependency Tree

the sentence can only be justified as a modifier of "apples" and it narrows down the set of apples from which John ate thus adding to the meaning of its head. Figure 1-2 shows the corresponding constituency tree, Note that this tree has 8 internal tree nodes in addition to

the leaf word nodes.

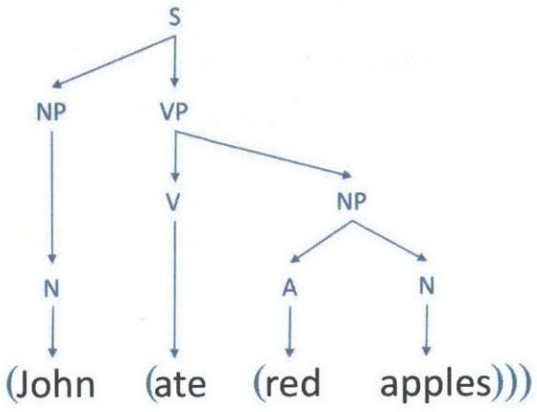


Figure 1-2: A Constituency Tree

The notion of head is not exclusive to dependency formalism, most other grammar formalisms, including constituency based ones, have some notion of head for spans or constituents. The question of how to identify the head of a constituent, or alternatively how to decide the direction of a binary dependency, is debatable. The head may differ depending on whether we are looking for semantic head or syntactic head. Semantic heads are usually the content words such as Nouns or Verbs, while syntactic heads maybe the function words such as Prepositions and Auxiliaries. However, even within syntactic theories, function words are not always marked as heads. For instance, in the dependencies involving Auxiliary and Verb, the Auxiliary word is usually considered head however in the dependencies between Nouns and Determiners, NP analysis (Noun Phrase where Noun is the head) is more common than DP analysis (Determiner Phrase where Determiner is the head). There are reasonable arguments to support both views. Therefore, different dependency based formalisms and even different dependency types within one formalism maybe based on different head finding principles. This lack of consensus in syntactic theories is partly the reason for inconsistency in different treebanks annotations.

In dependency based representation, modifiers are often further divided into two categories 1) the necessary modifiers also called "arguments" and 2) the optional modifiers, called "adjuncts". For instance, the subject and object modifiers of transitive Verbs are arguments, since a transitive Verb is incomplete in meaning without either subject or object; but Adjective modifiers of Nouns are not necessary and hence are adjuncts. In the example sentence, removing the object ("John ate") makes the sentence incomplete but removing the Adjective ("John ate apples") does not. Dependency parse may also include the labels of dependency links, for instance subject and object labels. However, the work presented in this thesis always assumes unlabeled trees.

One benefit of dependency representation, in contrast to constituency based representations, is its ability to handle discontinuities that frequently arise in free-word-order languages. This is because the dependency representation of a sentence is not tied to the surface order of words. However, majority of the NLP work on dependency parsing assumes projectivity which limits the ability of dependency representation. A projective dependency tree is the one for which the linear order of the words in sentence does not introduce any crossing dependency links. For instance, the tree in Figure 1-1 is projective (as is the case for most of English sentences), however the tree shown in Figure 1-3 is not projective. The

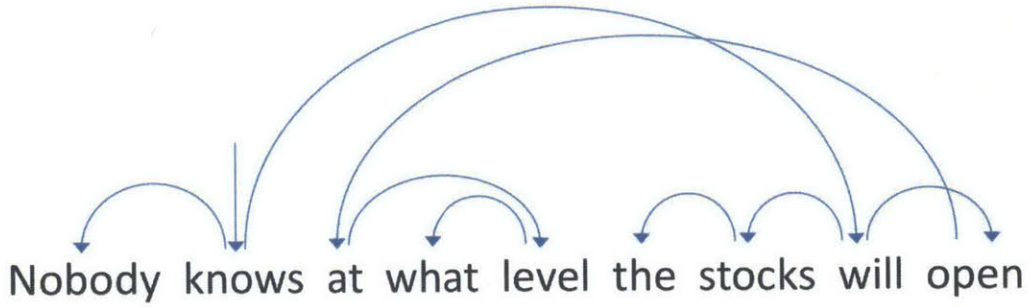


Figure 1-3: An example of a non-projective dependency tree

link between "open" and "at" crosses the link between "knows" and "will". This is a very uncommon occurrence for English but for other languages, such as German, non-projective

trees are very common. The work presented in this thesis assumes projectivity, however the basic ideas can be incorporated into a non-projective parser.

1.2 Dependency Parsing in NLP

Much like the linguistics theories, the dependency formalism gained popularity in NLP only in recent years. Initially, most of the parsing work was based on constituency formalism. However, comparative simplicity of dependency formalism, combined with the development of efficient algorithms [27], has made it the formalism of choice in many syntactic parsing works. Furthermore, the availability of CoNLL 06 and 07 datasets [14, 62] and the recent trends of multilingual evaluation of NLP tools has also contributed to the popularity of this framework for parsing research.

Early work in syntactic parsing (as well as in other areas of NLP) was focused primarily on English. Efforts were invested into developing annotated resources for English which could be used to train supervised parsing algorithms [53]. Availability of annotated resources allowed for designing parsing models that can capture detailed syntactic structures [22] such as sub-categorization frames and second order features. Moreover, the supervised training regime also allows for the use of feature rich discriminative training algorithms [57].

As the focus shifted towards languages other than English, the issue of lack of crosslingual portability of supervised parsers became apparent. Initial efforts to address this problem relied mainly on unsupervised parsing methods [47]. In contrast to their supervised counterparts, these methods simplified the parse representation to a bare-bones level: just brackets for constituency parse and just head-modifier tag pairs for dependency parse [47]. These simpler design choices were driven in part by the computational complexity of unsupervised learning algorithms and in part to enable the parser to identify consistent patterns in data. Since pattern identification is a lot easier if the sought after patterns are very high level. The issue of computational difficulty has been addressed in various works including the contrastive estimation work of Eisner et al. [70] and the sampling based algorithms

of Cohen et al. [10]. However, the issue of learning complex patterns from un-annotated data still remains. As mentioned earlier, the problem lies partly in difficulty of evaluating the patterns learned by an unsupervised parser. Evaluating unsupervised parsers against human annotations is bound to fail. In fact, for most unsupervised parsers, the undirected dependency accuracy is at least 10 points better than the directed dependency accuracy, this is not counting the ripple effect of those errors. In other words, like the linguists, the parser also quite often disagree with the annotations on the choice of head.

In past few years, efforts have been made to standardize the annotations across languages with a focus on enabling transfer of linguistic tools from resource-rich languages to resource-lean languages. One such work is the development of universal tagset by Petrov et al. [65]. By mapping part-of-speech tagsets used in different dependency treebanks to a universal coarse set, direct transfer of unlexicalized parser across languages is made possible [58]. However, the structural inconsistencies in annotations can not be handled by universal POS tag mapping. Furthermore, not all differences between treebanks of different languages are due to annotation differences. The languages *are* in fact different and direct transfer does not handle that, which makes it useful only when the source and target languages are structurally quite similar. Some of the work presented in this thesis is focused on crosslingual transfer when source and target languages are very different.

Another way of enhancing the performance of unsupervised parser is to introduce high-level weak supervision into the parser that guides it towards desired structures. This high level knowledge is typically incomplete and/or noisy. Examples include using partial HTML mark-up [77], partial semantic annotations [59], constraints induced from parallel data [33] and high-level rules [26]. Most of the work presented in this thesis is also closely related to this direction of research.

1.3 This thesis

We first present an example to motivate the use of linguistic knowledge. We then give a more detailed overview of different types of linguistic knowledge that we use followed by a high level account of techniques we employ to incorporate linguistic knowledge into parsing models. The last section summarizes major contributions of our work.

1.3.1 A Motivating Example

In dependency parsing formalism, for every word in the sentence, a decision is made about which other word in the sentence is the syntactic head of the word. While inducing these decisions, it has been standard practice to assume that the part-of-speech tag of each word is known. However, the connection between part-of-speech and the dependency structure, as defined in syntactic theories, is always ignored when learning dependency structures. For instance, take the example sentence shown in Figure 1-4. The dependency parse shown in the figure is produced by an unsupervised parser [47]. If we know that the part-of-speech tag of "the" is Article, then we also know, by definition, that "the Articles are the words that specify definiteness of Nouns". We can then eliminate many spurious attachments for the word "the". Note, however, that such information can only guide the parser, it can not completely disambiguate the decision. In the sentence in Figure 1-4 there are still multiple Noun candidates that can be modified by "the". Further information, such as the position of modifier with respect to the head and other modifiers of the head, maybe needed for total disambiguation. This calls for parsing models that can effectively make use of external linguistic knowledge while filling the information gaps via learning from the data.

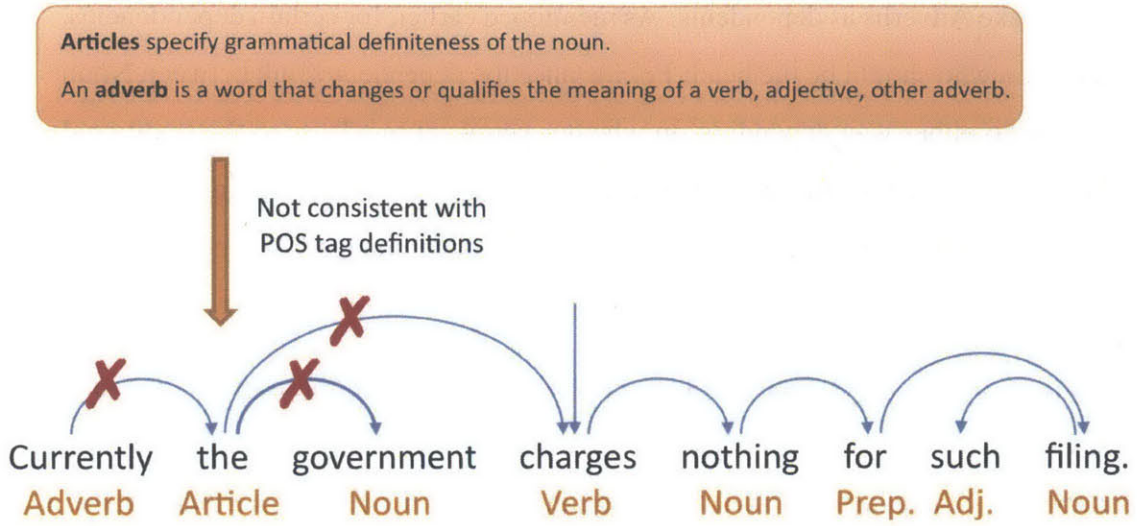


Figure 1-4: An example of how POS tag definitions may help eliminate wrong dependencies. The definitions of Article and Adverb disagree with wrong dependencies predicted by an unsupervised parser [47].

1.3.2 Forms of Linguistic Knowledge

When selecting which linguistic information to incorporate into our parsing models, we have two considerations in mind: First, the information should be either universal or easily available for most languages. Second, It should be representable in a form meaningful within the parsing framework. While the latter is a practical consideration, the former is intended to make our methods usable for most languages in the world.

Universal Rules

We first propose the use of a small set of universal dependency rules (see Table 1.1) as guiding linguistic information. These rules are universal because they are based on the definition of Part-Of-Speech (POS) categories. For example Adverbs are by definition the words that modify Verbs or Adjectives, so we add two rules stating that Verbs and

Adjectives can take Adverbs as dependents. As mentioned earlier, for certain dependencies, linguists differ on who should be the head. For instance the dependency link between Noun and Article can go either way depending on whether you want to adhere to the traditional view or the more recent transformational grammar. We chose the more commonly held traditional notion of Noun phrase. However, the rules can be changed depending on the desired annotation scheme.

We do not expect all the dependency relations to be consistent with these rules. Post-hoc analysis of CoNLL 2006-2007 shared task data [14, 62] reveals that the ratio of these rules varies greatly across languages; while most languages follow the rules more than 50% of the times, there are outliers too. For instance, Japanese dependencies are consistent with the rules only 35% of the times. Our method can handle this variability because we introduce the rules as soft constraints. In particular, we introduce them as expectation constraint over model posterior (discussed later in more detail).

Note that the universal rules only specify the head and the dependent tags, they do not specify their position/ordering in the sentence relative to each other. It is the absence of ordering information that makes them universal.

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

Table 1.1: The manually-specified universal dependency rules used in our experiments.

Linguistic Typology

Linguistic typology is a subfield within linguistics that studies systematic variations between languages; it characterizes the languages with respect to different aspects of structural variations. Examples include morphological (internal structure of words), phonetic (sound related), lexical (regarding the vocabulary of a language) or syntactic (word order) variations. We are interested in the characteristics concerning syntactic variations. An example of such variations would be the order of Subject, Verb and Object. English for instance is a SVO language i.e. the Subject precedes the Verb and the Object follows it, Urdu on the other hand is a VSO language.

We propose the use of syntactic typology of languages when transferring parsers trained on a set of source languages to a target language. This setup is of interest because syntactically annotated treebanks of reasonable sizes are available today for a couple of dozen languages. Annotation efforts for a new language can be saved if we can effectively transfer the information available in these treebanks. However, a new target language may share different aspects of its syntactic structure with different source languages. Therefore, a parser trained on a diverse set of source languages will not do well if transferred blindly. In this situation, the syntactic typological features can be useful. They can guide the parser to share the parameters of the target language only selectively with the source languages depending on shared typological characteristics.

Like the universal rules, typological information is also readily available for most languages via the online version of "The World Atlas of Language Structure" (WALS) [43]. In our experiments we only use those syntactic typology features that are defined in WALS for all the languages in our dataset (see Table 1.2). Note that the features listed in the Table 1.2 only specify the ordering of dependents with respect to the parent, the fact that undirected versions of those dependencies exist universally in all languages is implied. Our transfer

ID	Feature Description	Values
81A	Order of Subject, Object and Verb	SVO, SOV, VSO, VOS, OVS, OSV
85A	Order of Adposition and Noun	Postpositions, Prepositions, Inpositions
86A	Order of Genitive and Noun	Genitive-Noun, Noun-Genitive
87A	Order of Adjective and Noun	Adjective-Noun, Noun-Adjective
88A	Order of Demonstrative and Noun	Demonstrative-Noun, Noun-Demonstrative
89A	Order of Numeral and Noun	Numeral-Noun, Noun-Numeral

Table 1.2: The set of typological features that we use in our transfer based parsing model. For each feature, the first column gives the ID of the feature as used in WALS, the second column describes the feature and the last column enumerates the allowable values for the feature. Besides these values, each feature can also have a value of 'No dominant order'.

model also makes use of the universal nature of undirected dependencies.

Multiple Grammar Formalisms

While the work presented in this thesis is primarily based on dependency parsing framework, combining it with other frameworks, like constituency parsing, should help constrain the learning process. Incorporating constituency information is particularly easy if we limit the dependency structures to only projective trees. In a projective tree, the span projected by any node in the dependency tree forms a constituent. Consider for example the dependency tree in Figure 1-5 and its corresponding constituent brackets. Most of the brackets of the constituency tree are discernible via the dependency tree.

The nice thing about constituent brackets is the notion of substitutability i.e. the constituents of the same type should be usable interchangeably. This means that if we take an NP from one sentence and replace it with an NP in another sentence, the resulting sentence should still be grammatical. This ensures the context-free-ness of the constituency grammar. To incorporate the idea of substitutability into a dependency parser, we model distributions over the pairs of internal and external boundary tags of each constituent span. It

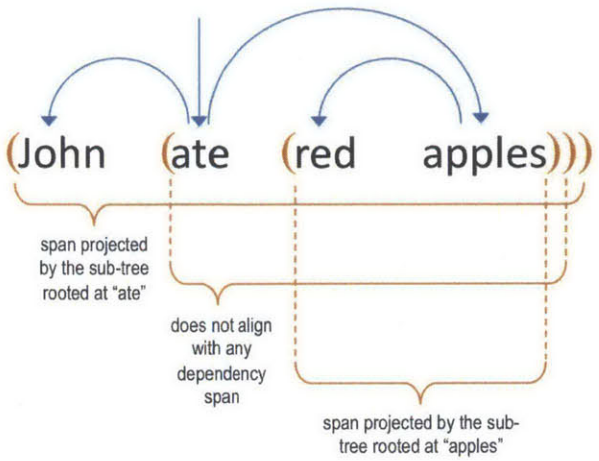


Figure 1-5: A dependency tree (represented by arrows) and corresponding constituency brackets (represented by parentheses)

is seemingly counter-intuitive to use context POS tags in order to model context-free-ness. However, explicitly modeling the context of constituents can be viewed as substitutability test.

Figure 1-6 shows an example of how constituency and context information may help correct common mistakes made by lightly supervised dependency parsers. The parse in the top is produced by a dependency based parser trained only on 10 annotated sentences. It erroneously predicts Verb "stepped" as parent of Preposition "to". The dependency parse at the bottom is produced by a parser that combines dependency and constituency views. This parser is also trained on the same 10 annotated sentences; Yet, the dependency tree produced by this parser does not have the erroneous dependency link. If we consider only local dependency decisions, then the first parse is much more probable than the second. This is because Verb is a very likely parent for Preposition. The true dependency, where the Preposition is headed by another Preposition, is relatively rare. It is not likely to be learned from a small set of training sentences. However, if we look at the constituent spans produced by the erroneous tree, they are very unlikely. For instance, based on the 10 an-

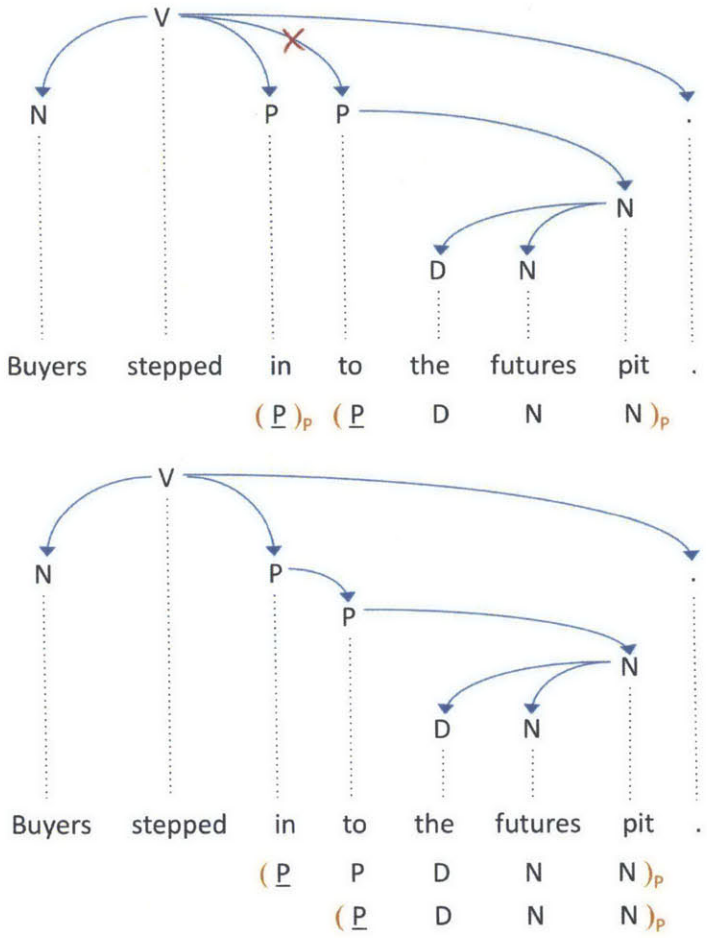


Figure 1-6: Trees produced by dependency-only parser (top) and combined dependency+constituency parser (bottom).

notated training sentences, having a preposition (in) that has only itself in its projected constituent span is not likely. Therefore, a combined parser has correctly predicted the attachment for the Preposition.

1.3.3 Techniques for Incorporating Linguistic Knowledge

Linguistic information can either be incorporated directly into the parsing model in the form of model parameters or it can be used to constrain the search space during training. The former is useful when incorporating relatively detailed linguistic concepts like constituency information or sub-categorization frames, the latter is more appropriate when linguistic knowledge is in the form incomplete high level information such as the universal linguistic rules discussed in the previous section.

Constrained Inference

Incorporating high-level linguistic rules directly into the parsing model can be challenging as it requires careful tuning of either the model structure or priors for each constraint. Instead, following the approach of [38], we constrain the posterior to satisfy the rules in expectation during inference. This effectively biases the inference toward linguistically plausible settings. We adapt this method to our Bayesian framework via variational approximation.

In standard variational inference, an intractable true posterior is approximated by a distribution from a tractable set [8]. This tractable set typically makes stronger independence assumptions between model parameters than the model itself. To incorporate the constraints, we further restrict the set to only include distributions that satisfy the specified expectation constraints over hidden variables.

In general, for some given model, let θ denote the entire set of model parameters and z and x denote the hidden structure and observations respectively. We are interested in estimating the posterior $p(\theta, z \mid x)$. Variational inference transforms this problem into an optimization problem where we try to find a distribution $q(\theta, z)$ from a restricted set $\mathcal{Q}$ that minimizes

the KL-divergence between $q(\theta, z)$ and $p(\theta, z \mid x)$. To make this minimization tractable, a mean field factorization is typically assumed. This means that q belongs to a set $\mathcal{Q}$ of distributions that factorize as follows:

$$q(\theta, z) = q(\theta)q(z).$$

We further constrain q to be from the subset of $\mathcal{Q}$ that satisfies the expectation constraint $E_q[f(z)] \leq b$ where f is a deterministically computable function of the hidden structures. In our model, f counts the dependency edges that are an instance of one of the declaratively specified dependency rules (1.1), while b is the proportion of the total dependencies that we expect should fulfill this constraint (set to 0.8 in our experiments).

Findings: We test the effectiveness of our grammar induction model on six Indo-European languages from three language groups: English, Danish, Portuguese, Slovene, Spanish, and Swedish. Though these languages share a high-level Indo-European ancestry, they cover a diverse range of syntactic phenomenon. Our results demonstrate that universal rules greatly improve the accuracy of dependency parsing across all of these languages, outperforming the state-of-the-art unsupervised grammar induction methods [44, 7].

Linguistically Motivated Model Design

One obvious way of incorporating linguistic knowledge is via model structure. A linguistically motivated model design can either learn the desired information from the data or can make use of known information in the form of observed parameters. However, the challenge lies in capturing important pieces of information without overly complicating the model. Too much detail will cause sparsity and/or intractability issues, too little will

miss on important patterns. We address this challenge first by focusing on the phenomena that are salient to the setup at hand and then by introducing corpus level models that allow different linguistic views in a single model without making inference intractable.

Model Design for Parser Transfer We first introduce a parsing model that is designed specifically for a cross-lingual transfer setup. In this setup, treebank annotations are available for a set of source languages. The goal is to effectively transfer this information to a new target language for which language specific treebank data is not available. The idea of parser transfer across languages implicitly relies on universal aspects of syntactic structure. However, existing work in this direction does not make explicit use of those aspects.

We propose a parser that explicitly separates universal components from the language specific components. This approach is rooted in linguistic theory that characterizes the connection between languages at various levels of sharing. Some syntactic properties are universal across languages. For instance, Nouns take Adjectives and Articles as dependents, but not Adverbs. However, the order of these dependents with respect to the parent is influenced by the typological features of each language.

We capture this intuition via a two-tier model that separates the *selection* of dependents from their *ordering*: *Selection Component* determines the dependent tags given the parent tag. *Ordering Component* determines the position of each dependent tag with respect to its parent (right or left) and the order within the right and left dependents. This factorization constitutes a departure from traditional parsing models where these decisions are tightly coupled. By separating the two, the model is able to support different degrees of cross-lingual sharing on each level.

The parameters of the selection component are assumed to be universal and can therefore be borrowed from any or all of the source languages. Note that the universal component of

this model can be viewed as an alternative for the undirected universal rules declaratively specified in the rules based parser from previous section.

For ordering component, we allow partial sharing based on the typological features listed in Table 1.2 in Section 1.3.2. In particular, we define ordering distribution as a log-linear model whose dependence on any particular language is captured only via typological feature, i.e. the identity of the language is never used as a feature.

$$P(d|a,h,l) = \frac{1}{Z(a,h,l)} e^{w \cdot g(d,a,h,v_l)}$$

Where a and h are the dependent and the head tags, l is the language, d is the orientation of the dependent with respect to the head (left or right). w are the feature weights and g is the feature function. Note that g has access to the language l only via its typological features v_l. Thus even if the target language is not closely related to any single source language, its

ID	English	Portuguese	Arabic
81A	SVO	SVO	VSO
85A	Preposition	Preposition	Preposition
87A	Adjective-Noun	Noun-Adjective	Noun-Adjective
88A	Demonstrative-Noun	Demonstrative-Noun	Demonstrative-Noun

Table 1.3: Typological word-order features of English, Portuguese and Arabic.

ordering decisions are selectively informed by different languages it shares features with. For instance, Portuguese is a prepositional language like English, but the order of its Noun-Adjective dependency is different from English and matches that of Arabic (see Table 1.3). The typological features of Portuguese enable the sharing of right parameters with each language and block the irrelevant parameters.

Findings: We evaluated our selective sharing model on 17 languages from 10 language families. On this diverse set, our model consistently outperforms state-of-the-art multi-

lingual dependency parsers. Performance gain, averaged over all the languages, is 5.9% when compared to the highest baseline. Our model achieves the most significant gains on non-Indo-European languages, where we see a 14.4% improvement.

Corpus Level Models with Multiple Linguistic Views Combining multiple ways of solving the same task rarely fails. This is particularly true when models are simple, misspecified, and capture largely complementary aspects of the data. The insight has been used frequently in unsupervised parsing. For example, early successes [47] sought to combine dependency and constituent parsing. Another perspective to combining models is to incorporate declarative knowledge, insufficient on its own, into an otherwise unsupervised model [21, 26, 60].

In NLP, for unsupervised tasks, generative models have frequently been the framework of choice. The locally normalized parameters in generative models eliminate the need to computing the intractable normalization term. However, incorporating multiple overlapping views into a generative model usually makes the inference computationally intractable. One common mechanism for handling multiple views in a generative framework is the product of experts [47, 72, 13]. In this scenario, each view forms a generative model and the latent structure is scored using the product of their probabilities. These models are simple and effective in practice. However, product of experts models typically require some form of approximation during inference.

We propose unsupervised Bayesian models of dependency parsing that operate on the corpus level. By dispensing with the per-sentence view of modeling, we can easily merge multiple ways of scoring or constraining parsing decisions. We begin with a simple generative dependency parsing model, akin to [47], and adorn it with various complementary views of scoring. Different views are incorporated as additional parameters that pertain to the same parsing decisions. By integrating over the parameters (i.e., exploring jointly opti-

mal parameter settings), we necessarily couple parsing decisions across the sentences in the corpus. It is still possible, however, to sample a new improved parse for each sentence, one sentence at a time, in a (collapsed) Gibbs' sampling framework, or approximately using Metropolis-Hastings. We experiment with several such alternative views.

Findings: We evaluate our method on 19 languages, using dependency data from CONLL 2007 and CONLL 2006 datasets. For unsupervised experiments, we compare our results against the state-of-the-art parser [76] which combines several parsing models. On average, our model outperforms this baseline by 1.9%.

1.3.4 Contributions

The contribution of this work is threefold:

- We introduce the notion of a **universal view of dependency parsing**, first via universal rules and then by separating universal selection component from language specific ordering component for parser transfer. Our experiments show that this two tier-approach is quite effective for implementing language independent parsers.

- We are the first to exploit **linguistic typology** to model crosslingual syntactic variations. We thus enable the transfer of ordering information from a diverse set of source languages to an un-related target language. We have shown that even very high-level typological information can help make the parsers more portable across languages.

- We propose a **corpus-level Bayesian framework** that makes it easy to merge multiple overlapping views of data into one model. The framework is generic and can be used for tasks other than dependency parsing. We demonstrate the effectiveness of this approach by combining dependency and constituency views in one parser, yielding significant performance gains on dependency predictions. Existing research in NLP has demonstrated the benefits of joint learning of multiple tasks [79, 32], this framework can also be used as a mechanism for joint learning.

1.4 Outline

The remainder of this thesis is organized as follows:

- **Chapter 2** describes in detail our rule-based parsing method. This method uses universal dependency rules to constrain parameter search for an otherwise unsupervised parser.
- **Chapter 3** presents our parsing model designed specifically for crosslingual parser transfer. This parser makes use of linguistic typology to selectively transfer parsing information to a target language according to its syntactic connections with source languages.
- **Chapter 4** proposes a corpus-level Bayesian framework for unsupervised parsing. This method allows to incorporate overlapping linguistic views into one model without making inference intractable.
- **Chapter 5** summarizes major findings and points to directions for future research.

Chapter 2

Using Universal Linguistic Rules for Grammar Induction

Despite surface differences, human languages exhibit striking similarities in many fundamental aspects of syntactic structure. These structural correspondences, referred to as *syntactic universals*, have been extensively studied in linguistics [1, 16, 81, 61] and underlie many approaches in multilingual parsing. In fact, much recent work has demonstrated that learning cross-lingual correspondences from corpus data greatly reduces the ambiguity inherent in syntactic analysis [49, 15, 20, 71, 7].

In this chapter, we present a grammar induction approach that exploits these structural correspondences by declaratively encoding a small set of universal dependency rules. As input to the model, we assume a corpus annotated with coarse syntactic categories (i.e., high-level part-of-speech tags) and a set of universal rules defined over these categories, such as those in Table 2.1. These rules incorporate the definitional properties of syntactic categories in terms of their interdependencies and thus are universal across languages. They can potentially help disambiguate structural ambiguities that are difficult to learn from data

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	

Table 2.1: The manually-specified universal dependency rules used in our experiments. These rules specify head-dependent relationships between coarse (i.e., unsplit) syntactic categories. An explanation of the ruleset is provided in Section 2.4.

alone — for example, our rules prefer analyses in which verbs are dependents of auxiliaries, even though analyzing auxiliaries as dependents of verbs is also consistent with the data. Leveraging these universal rules has the potential to improve parsing performance for a large number of human languages; this is particularly relevant to the processing of low-resource languages. Furthermore, these universal rules are compact and well-understood, making them easy to manually construct.

In addition to these universal dependencies, each specific language typically possesses its own idiosyncratic set of dependencies. We address this challenge by requiring the universal constraints to only hold in expectation rather than absolutely, i.e., we permit a certain number of violations of the constraints.

We formulate a generative Bayesian model that explains the observed data while accounting for declarative linguistic rules during inference. These rules are used as expectation constraints on the posterior distribution over dependency structures. This approach is based on the posterior regularization technique [37], which we apply to a variational inference algorithm for our parsing model. Our model can also optionally refine common high-level syntactic categories into per-language categories by inducing a clustering of words using Dirichlet Processes [30]. Since the universals guide induction toward linguistically plau-

sible structures, automatic refinement becomes feasible even in the absence of manually annotated syntactic trees.

We test the effectiveness of our grammar induction model on six Indo-European languages from three language groups: English, Danish, Portuguese, Slovene, Spanish, and Swedish. Though these languages share a high-level Indo-European ancestry, they cover a diverse range of syntactic phenomenon. Our results demonstrate that universal rules greatly improve the accuracy of dependency parsing across all of these languages, outperforming current state-of-the-art unsupervised grammar induction methods [44, 7].

2.1 Related Work

Learning with Linguistic Constraints Our work is situated within a broader class of unsupervised approaches that employ declarative knowledge to improve learning of linguistic structure [41, 17, 38, 21, 26, 50]. The way we apply constraints is closest to the latter two approaches of posterior regularization and generalized expectation criteria.

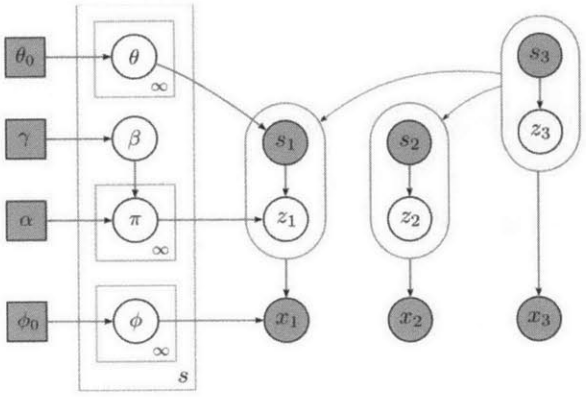
In the posterior regularization framework, constraints are expressed in the form of expectations on posteriors [38, 33, 37, 34]. This design enables the model to reflect constraints that are difficult to encode via the model structure or as priors on its parameters. In their approach, parameters are estimated using a modified EM algorithm, where the E-step minimizes the KL-divergence between the model posterior and the set of distributions that satisfies the constraints. Our approach also expresses constraints as expectations on the posterior; we utilize the machinery of their framework within a variational inference algorithm with a mean field approximation.

Generalized expectation criteria, another technique for declaratively specifying expectation constraints, has previously been successfully applied to the task of dependency pars-

ing [26]. This objective expresses constraints in the form of preferences over model expectations. The objective is penalized by the square distance between model expectations and the prespecified values of the expectation. This approach yields significant gains compared to a fully unsupervised counterpart. The constraints they studied are corpus- and language-specific. Our work demonstrates that a small set of language-independent universals can also serve as effective constraints. Furthermore, we find that our method outperforms the generalized expectation approach using corpus-specific constraints.

Learning to Refine Syntactic Categories Recent research has demonstrated the usefulness of automatically refining the granularity of syntactic categories. While most of the existing approaches are implemented in the supervised setting [31, 66], [52] propose a non-parametric Bayesian model that learns the granularity of PCFG categories in an unsupervised fashion. For each non-terminal grammar symbol, the model posits a Hierarchical Dirichlet Process over its refinements (subsymbols) to automatically learn the granularity of syntactic categories. As with their work, we also use non-parametric priors for category refinement and employ variational methods for inference. However, our goal is to apply category refinement to dependency parsing, rather than to PCFGs, requiring a substantially different model formulation. While [52] demonstrated empirical gains on a synthetic corpus, our experiments focus on unsupervised category refinement on real language data.

Universal Rules in NLP Despite the recent surge of interest in multilingual learning [49, 20, 71, 7], there is surprisingly little computational work on linguistic universals. On the acquisition side, [25] proposed a computational technique for discovering universal implications in typological features. More closely related to our work is the position paper by [3], which advocates the use of manually-encoded cross-lingual generalizations for the development of NLP systems. She argues that a system employing such knowledge could



s	-	coarse symbol (observed)
z	-	refined subsymbol
x	-	word (observed)
θ_{szc}	-	distr over child coarse symbols for each parent s and z and context c
β_s	-	top-level distr over sub-symbols for s
$\pi_{ss'z'c}$	-	distr over subsymbols for each s, parent s' and z', and context c
ϕ_{sz}	-	distr over words for s and z

Figure 2-1: Graphical representation of the model and a summary of the notation. There is a copy of the outer plate for each distinct symbol in the observed coarse tags. Here, node 3 is shown to be the parent of nodes 1 and 2. Shaded variables are observed, square variables are hyperparameters. The elongated oval around s and z represents the two variables jointly. For clarity the diagram omits some arrows from θ to each s, π to each z, and ϕ to each x.

be easily adapted to a particular language by specializing this high level knowledge based on the typological features of the language. We also argue that cross-language universals are beneficial for automatic language processing; however, our focus is on learning language-specific adaptations of these rules from data.

2.2 Model

The central hypothesis of this work is that unsupervised dependency grammar induction can be improved using universal linguistic knowledge. Toward this end our approach is comprised of two components: a probabilistic model that explains how sentences are generated from latent dependency structures and a technique for incorporating declarative rules into the inference process.

We first describe the generative story in this section before turning to how constraints are applied during inference in Section 2.3. Our model takes as input (i.e., as observed) a set

For each observed coarse symbol t:

1. Draw top-level infinite multinomial over subsymbols $\beta_t \sim \text{GEM}(\gamma)$.
2. For each subsymbol s of symbol t:
   (a) Draw word emission multinomial $\phi_{ts} \sim \text{Dir}(\phi_0)$.
   (b) For each context value c:
      i. Draw child symbol generation multinomial $\theta_{tsc} \sim \text{Dir}(\theta_0)$.
      ii. For each child symbol t':
         A. Draw second-level infinite multinomial over subsymbols $\pi_{t'tsc} \sim \text{DP}(\alpha, \beta_{t'})$.

For each tree node i generated in context c by parent symbol t' and parent subsymbol s':

1. Draw coarse symbol $t_i \sim \text{Mult}(\theta_{t's'c})$.
2. Draw subsymbol $s_i \sim \text{Mult}(\pi_{t_it's'c})$.
3. Draw word $x_i \sim \text{Mult}(\phi_{s_it_i})$.

Table 2.2: The generative process for model parameters and parses. In the above GEM, DP, Dir, and Mult refer respectively to the stick breaking distribution, Dirichlet process, Dirichlet distribution, and multinomial distribution.

of sentences where each word is annotated with a coarse part-of-speech tag. Table 2.2 provides a detailed technical description of our model's generative process, and Figure 2-1 presents a model diagram.

Generating Symbols and Words We describe how a single node of the tree is generated before discussing how the entire tree structure is formed. Each node of the dependency tree is comprised of three random variables: an observed coarse symbol t, a hidden refined subsymbol s, and an observed word x. In the following let the parent of the current node have symbol t' and subsymbol s'; the root node is generated from separate root-specific

distributions. Subsymbol refinement is an optional component of the full model and can be omitted by deterministically equating t and s. As we explain at the end of this section, without this aspect the generative story closely resembles the classic dependency model with valence (DMV) of [47].

First we draw symbol t from a finite multinomial distribution with parameters $\theta_{t's'c}$. As the indices indicate, we have one such set of multinomial parameters for every combination of parent symbol t' and subsymbol s' along with a *context* c. Here the context of the current node can take one of six values corresponding to every combination of direction (left or right) and valence (first, second, or third or higher child) with respect to its parent. The prior (base distribution) for each $\theta_{t's'c}$ is a symmetric Dirichlet with hyperparameter θ_0.

Next we draw the refined syntactic category subsymbol s from an infinite multinomial with parameters $\pi_{tt's'c}$. Here the selection of π is indexed by the current node's coarse symbol t, the symbol t' and subsymbol s' of the parent node, and the context c of the current node with respect to its parent.

For each unique coarse symbol t we tie together the distributions $\pi_{tt's'c}$ for all possible parent and context combinations (i.e., t', s', and c) using a Hierarchical Dirichlet Process (HDP). Specifically, for a single t, each distribution $\pi_{tt's'c}$ over subsymbols is drawn from a DP with concentration parameter α and base distribution β_t over subsymbols. This base distribution β_t is itself drawn from a GEM prior with concentration parameter γ. By formulating the generation of s as an HDP, we can share parameters for a single coarse symbol's subsymbol distribution while allowing for individual variability based on node parent and context. Note that parameters are not shared across different coarse symbols, preserving the distinctions expressed via the coarse tag annotations.

Finally, we generate the word x from a finite multinomial with parameters ϕ_{sz}, where s and z are the symbol and subsymbol of the current node. The ϕ distributions are drawn from a

symmetric Dirichlet prior.

Generating the Tree Structure We now consider how the structure of the tree arises. We follow an approach similar to the widely-referenced DMV model [47], which forms the basis of the current state-of-the-art unsupervised grammar induction model [44]. After a node is drawn we generate children on each side until we produce a designated STOP symbol. We encode more detailed valence information than [47] and condition child generation on parent valence. Specifically, after drawing a node we first decide whether to proceed to generate a child or to stop conditioned on the parent symbol and subsymbol and the current context (direction and valence). If we decide to generate a child we follow the previously described process for constructing a node. We can combine the stopping decision with the generation of the child symbol by including a distinguished STOP symbol as a possible outcome in distribution θ.

No-Split Model Variant In the absence of subsymbol refinement (i.e., when subsymbol s is set to be identical to coarse symbol t), our model simplifies in some respects. In particular, the HDP generation of s is obviated and word x is drawn from a word distribution ϕ_t indexed solely by coarse symbol t. Since both t and x are observed, the actual value of ϕ has no impact on the learning process. The resulting simplified model closely resembles DMV [47], except that it 1) encodes richer context and valence information, and 2) imposes a Dirichlet prior on the symbol distribution θ.

2.3 Inference with Constraints

We now describe how to augment our generative model of dependency structure with constraints derived from linguistic knowledge. Incorporating arbitrary linguistic rules directly

in the generative story is challenging as it requires careful tuning of either the model structure or priors for each constraint. Instead, following the approach of [38], we constrain the posterior to satisfy the rules in expectation during inference. This effectively biases the inference toward linguistically plausible settings.

In standard variational inference, an intractable true posterior is approximated by a distribution from a tractable set [8]. This tractable set typically makes stronger independence assumptions between model parameters than the model itself. To incorporate the constraints, we further restrict the set to only include distributions that satisfy the specified expectation constraints over hidden variables.

In general, for some given model, let θ denote the entire set of model parameters and z and x denote the hidden structure and observations respectively. We are interested in estimating the posterior $p(\theta, z \mid x)$. Variational inference transforms this problem into an optimization problem where we try to find a distribution $q(\theta, z)$ from a restricted set $\mathcal{Q}$ that minimizes the KL-divergence between $q(\theta, z)$ and $p(\theta, z \mid x)$:

$$\mathrm{KL}(q(\theta, z) \parallel p(\theta, z \mid x)) = \int q(\theta, z) \log \frac{q(\theta, z)}{p(\theta, z, x)} d\theta dz + \log p(x).$$

Rearranging the above yields:

$$\log p(x) = \mathrm{KL}(q(\theta, z) \parallel p(\theta, z \mid x)) + \mathcal{F},$$

where $\mathcal{F}$ is defined as

$$\mathcal{F} \equiv \int q(\theta, z) \log \frac{p(\theta, z, x)}{q(\theta, z)} d\theta dz. \tag{2.1}$$

Thus $\mathcal{F}$ is a lower bound on likelihood. Maximizing this lower bound is equivalent to minimizing the KL-divergence between $p(\theta, z \mid x)$ and $q(\theta, z)$. To make this maximization tractable we make a mean field assumption that q belongs to a set $\mathcal{Q}$ of distributions that

factorize as follows:

$$q(\theta, z) = q(\theta)q(z).$$

We further constrain q to be from the subset of $\mathcal{Q}$ that satisfies the expectation constraint $E_q[f(z)] \leq b$ where f is a deterministically computable function of the hidden structures. In our model, for example, f counts the dependency edges that are an instance of one of the declaratively specified dependency rules, while b is the proportion of the total dependencies that we expect should fulfill this constraint.[1]

With the mean field factorization and the expectation constraints in place, solving the maximization of $\mathcal{F}$ in (2.1) separately for each factor yields the following updates:

$$q(\theta) = \underset{q(\theta)}{\operatorname{argmin}} \operatorname{KL}\left(q(\theta) \parallel q'(\theta)\right), \tag{2.2}$$

$$q(z) = \underset{q(z)}{\operatorname{argmin}} \operatorname{KL}\left(q(z) \parallel q'(z)\right) \quad s.t. \quad E_{q(z)}[f(z)] \leq b, \tag{2.3}$$

where

$$q'(\theta) \propto \exp E_{q(z)}[\log p(\theta, z, x)], \tag{2.4}$$

$$q'(z) \propto \exp E_{q(\theta)}[\log p(\theta, z, x)]. \tag{2.5}$$

We can solve (2.2) by setting $q(\theta)$ to $q'(\theta)$ — since $q(z)$ is held fixed while updating $q(\theta)$, the expectation function of the constraint remains constant during this update. As shown by [38], the update in (2.3) is a constrained optimization problem and can be solved by performing gradient search on its dual:

$$\underset{\lambda}{\operatorname{argmin}} \lambda^\top b + \log \sum_z q'(z) \exp(-\lambda^\top f(z)) \tag{2.6}$$

[1] Constraints of the form $E_q[f(z)] \geq b$ are easily imposed by negating $f(z)$ and b.

For a fixed value of λ the optimal $q(z) \propto q'(z)\exp(-\lambda^\top f(z))$. By updating $q(\theta)$ and $q(z)$ as in (2.2) and (2.3) we are effectively maximizing the lower bound $\mathcal{F}$.

2.3.1 Variational Updates

We now derive the specific variational updates for our dependency induction model. First we assume the following mean-field factorization of our variational distribution:

$$q(\beta, \theta, \pi, \phi, z) = q(z) \cdot \prod_{t'} q(\beta_{t'}) \cdot \prod_{s'=1}^{T} q(\phi_{t's'}) \cdot \prod_{c} q(\theta_{t's'c}) \cdot \prod_{t} q(\pi_{tt's'c}), \tag{2.7}$$

where t' varies over the set of unique symbols in the observed tags, s' denotes subsymbols for each symbol, c varies over context values comprising a pair of direction (left or right) and valence (first, second, or third or higher) values, and t corresponds to child symbols. z refers to all latent variables, including dependency trees and subsymbols.

We restrict $q(\theta_{t's'c})$ and $q(\phi_{t's'})$ to be Dirichlet distributions and $q(z)$ to be multinomial. As with prior work [51], we assume a degenerate $q(\beta) \equiv \delta_{\beta^*}(\beta)$ for tractability reasons, i.e., all mass is concentrated on some single β^*. We also assume that the top level stick-breaking distribution is truncated at T, i.e., $q(\beta)$ assigns zero probability to integers greater than T. Because of the truncation of β, we can approximate $q(\pi_{tt's'c})$ with an asymmetric finite dimensional Dirichlet.

The factors are updated one at a time holding all other factors fixed. The variational update for $q(\pi)$ is given by:

$$q(\pi_{tt's'c}) = \text{Dir}\left(\pi_{tt's'c}; \alpha\beta_t + E_{q(z)}[C_{tt's'c}(z)]\right),$$

where term $E_{q(z)}[C_{tt's'c}(z)]$ is the expected count w.r.t. $q(z)$ of child symbol t and subsym-

bol s in context c when generated by parent symbol t' and subsymbol s'. See Appendix B.2 for the derivation of this update.

Similarly, the updates for $q(\theta)$ and $q(\phi)$ are given by:

$$q(\theta_{t's'c}) = \text{Dir}\left(\theta_{t's'c}; \theta_0 + E_{q(z)}[C_{t's'c}(s)]\right),$$
$$q(\phi_{t's'}) = \text{Dir}\left(\phi_{t's'}; \phi_0 + E_{q(z)}[C_{t's'}(x)]\right),$$

where $C_{t's'c}(s)$ is the count of child symbol t being generated by the parent symbol t' and subsymbol s' in context c and $C_{t's'x}$ is the count of word x being generated by symbol t' and subsymbol s' (see Appendix B.1 for details).

The only factor affected by the expectation constraints is $q(z)$. Recall from the previous section that the update for $q(z)$ is performed via gradient search on the dual of a constrained minimization problem of the form:

$$q(z) = \underset{q(z)}{\text{argmin}}\, \text{KL}(q(z) \parallel q'(z)).$$

Thus we first compute the update for $q'(z)$ (see derivation in Appendix B.3):

$$\begin{aligned} q'(z) \propto \prod_{n=1}^{N} \prod_{j=1}^{len(n)} & (\exp E_{q(\phi)}[\log \phi_{t_{nj}s_{nj}}(x_{nj})] \\ & \times \exp E_{q(\theta)}[\log \theta_{t_{h(nj)}s_{h(nj)}c_{nj}}(t_{nj})] \\ & \times \exp E_{q(\pi)}[\log \pi_{t_{nj}t_{h(nj)}s_{h(nj)}c_{nj}}(s_{nj})]), \end{aligned}$$

where N is the total number of sentences, $len(n)$ is the length of sentence n, and index $h(nj)$ refers to the head of the jth node of sentence n. Given this $q'(z)$ a gradient search is performed using (2.6) to find the optimal λ and thus the primal solution for updating $q(z)$.

1. Identify non-recursive NPs:
   - All nouns, pronouns and possessive marker are part of an NP.
   - All adjectives, conjunctions and determiners immediately preceding an NP are part of the NP.
2. The first verb or modal in the sentence is the headword.
3. All words in an NP are headed by the last word in the NP.
4. The last word in an NP is headed by the word immediately before the NP if it is a preposition, otherwise it is headed by the headword of the sentence if the NP is before the headword, else it is headed by the word preceding the NP.
5. For the first word set its head to be the headword of the sentence. For each other word set its headword to be the previous word.

Table 2.3: English-specific dependency rules.

Finally, we update the degenerate factor $q(\beta_s)$ with the projected gradient search algorithm used by [51].

2.4 Linguistic Constraints

Universal Dependency Rules We compile a set of 13 universal dependency rules consistent with various linguistic accounts [16, 61], shown in Table 2.1. These rules are defined over coarse part-of-speech tags: Noun, Verb, Adjective, Adverb, Pronoun, Article, Auxiliary, Preposition, Numeral and Conjunction. Each rule specifies a part-of-speech for the head and argument but does not provide ordering information.

We require that a minimum proportion of the posterior dependencies be instances of these

rules in expectation. In contrast to prior work on rule-driven dependency induction [26], where each rule has a separately specified expectation, we only set a single minimum expectation for the proportion of all dependencies that must match one of the rules. This setup is more relevant for learning with universals since individual rule frequencies vary greatly between languages.

English-specific Dependency Rules For English, we also consider a small set of hand-crafted dependency rules designed by Michael Collins[2] for deterministic parsing, shown in Table 2.3. Unlike the universals from Table 2.1, these rules alone are enough to construct a full dependency tree. Thus they allow us to judge whether the model is able to improve upon a human-engineered deterministic parser. Moreover, with this dataset we can assess the additional benefit of using rules tailored to an individual language as opposed to universal rules.

2.5 Experimental Setup

Datasets and Evaluation We test the effectiveness of our grammar induction approach on English, Danish, Portuguese, Slovene, Spanish, and Swedish. For English we use the Penn Treebank [53], transformed from CFG parses into dependencies with the Collins head finding rules [22]; for the other languages we use data from the 2006 CoNLL-X Shared Task [14]. Each dataset provides manually annotated part-of-speech tags that are used for both training and testing. For comparison purposes with previous work, we limit the cross-lingual experiments to sentences of length 10 or less (not counting punctuation). For English, we also explore sentences of length up to 20.

[2]Personal communication.

The final output metric is directed dependency accuracy. This is computed based on the Viterbi parses produced using the final unnormalized variational distribution $q(z)$ over dependency structures.

Hyperparameters and Training Regimes Unless otherwise stated, in experiments with rule-based constraints the expected proportion of dependencies that must satisfy those constraints is set to 0.8. This threshold value was chosen based on minimal tuning on a single language and ruleset (English with universal rules) and carried over to each other experimental condition. A more detailed discussion of the threshold's empirical impact is presented in Section 2.6.1.

Variational approximations to the HDP are truncated at 10. All hyperparameter values are fixed to 1 except α which is fixed to 10.

We also conduct a set of *No-Split* experiments to evaluate the importance of syntactic refinement; in these experiments each coarse symbol corresponds to only one refined symbol. This is easily effected during inference by setting the HDP variational approximation truncation level to one.

For each experiment we run 50 iterations of variational updates; for each iteration we perform five steps of gradient search to compute the update for the variational distribution $q(z)$ over dependency structures.

2.6 Results

In the following section we present our primary cross-lingual results using universal rules (Section 2.6.1) before performing a more in-depth analysis of model properties such as sensitivity to ruleset selection and inference stability (Section 2.6.2).

	DMV	PGI	No-Split	HDP-DEP
English	47.1	62.3	71.5	**71.9** (0.3)
Danish	33.5	41.6	48.8	**51.9** (1.6)
Portuguese	38.5	63.0	54.0	**71.5** (0.5)
Slovene	38.5	48.4	50.6	**50.9** (5.5)
Spanish	28.0	58.4	64.8	**67.2** (0.4)
Swedish	45.3	58.3	**63.3**	62.1 (0.5)

Table 2.4: Directed dependency accuracy using our model with universal dependency rules (No-Split and HDP-DEP), compared to DMV [47] and PGI [7]. The DMV results are taken from [7]. Bold numbers indicate the best result for each language. For the full model, the standard deviation in performance over five runs is indicated in parentheses.

2.6.1 Main Cross-Lingual Results

Table 2.4 shows the performance of both our full model (HDP-DEP) and its No-Split version using universal dependency rules across six languages. We also provide the performance of two baselines — the dependency model with valence (DMV) [47] and the phylogenetic grammar induction (PGI) model [7].

HDP-DEP outperforms both DMV and PGI across all six languages. Against DMV we achieve an average absolute improvement of 24.1%. This improvement is expected given that DMV does not have access to the additional information provided through the universal rules. PGI is more relevant as a point of comparison, since it is able to leverage multilingual data to learn information similar to what we have declaratively specified using universal rules. Specifically, PGI reduces induction ambiguity by connecting language-specific parameters via phylogenetic priors. We find, however, that we outperform PGI by an average margin of 7.2%, demonstrating the benefits of explicit rule specification.

An additional point of comparison is the lexicalized unsupervised parser of [44], which yields the current state-of-the-art unsupervised accuracy on English at 68.8%. Our method also outperforms this approach, without employing lexicalization and sophisticated smooth-

English			
Rule Excluded	Acc	Loss	Gold Freq
Preposition → Noun	61.0	10.9	5.1
Verb → Noun	61.4	10.5	14.8
Noun → Noun	64.4	7.5	10.7
Noun → Article	64.7	7.2	8.5
Spanish			
Rule Excluded	Acc	Loss	Gold Freq
Preposition → Noun	53.4	13.8	8.2
Verb → Noun	61.9	5.4	12.9
Noun → Noun	62.6	4.7	2.0
Root → Verb	65.4	1.8	12.3

Table 2.5: Ablation experiment results for universal dependency rules on English and Spanish. For each rule, we evaluate the model using the ruleset excluding that rule, and list the most significant rules for each language. The second last column is the absolute loss in performance compared to the setting where all rules are available. The last column shows the percentage of the gold dependencies that satisfy the rule.

ing as they do. This result suggests that combining the complementary strengths of their approach and ours can yield further performance improvements.

Table 2.4 also shows the *No-Split* results where syntactic categories are not refined. We find that such refinement usually proves to be beneficial, yielding an average performance gain of 3.7%. However, we note that the impact of incorporating splitting varies significantly across languages. Further understanding of this connection is an area of future research.

Finally, we note that our model exhibits low variance for most languages. This result attests to how the expectation constraints consistently guide inference toward high-accuracy areas of the search space.

Ablation Analysis Our next experiment seeks to understand the relative importance of the various universal rules from Table 2.1. We study how accuracy is affected when each

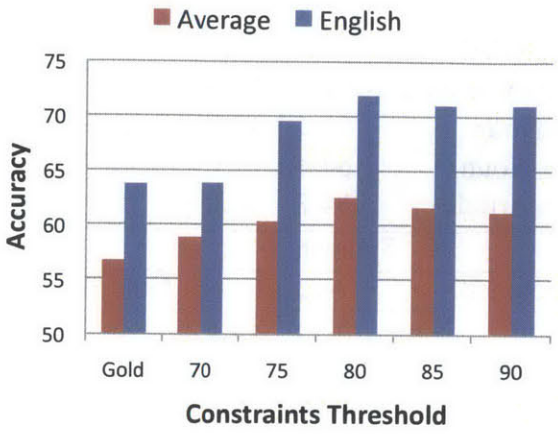


Figure 2-2: Accuracy of our model with different threshold settings, on English only and averaged over all languages. "Gold" refers to the setting where each language's threshold is set independently to the proportion of gold dependencies satisfying the rules — for English this proportion is 70%, while the average proportion across languages is 63%.

of the rules is removed one at a time for English and Spanish. Table 2.5 lists the rules with the greatest impact on performance when removed. We note the high overlap between the most significant rules for English and Spanish.

We also observe that the relationship between a rule's frequency and its importance for high accuracy is not straightforward. For example, the "Preposition $\rightarrow$ Noun" rule, whose removal degrades accuracy the most for both English and Spanish, is not the most frequent rule in either language. This result suggests that some rules are harder to learn than others regardless of their frequency, so their presence in the specified ruleset yields stronger performance gains.

Varying the Constraint Threshold In our main experiments we require that at least 80% of the expected dependencies satisfy the rule constraints. We arrived at this threshold by tuning on the basis of English only. As shown in Figure 2-2, for English a broad band of

		Length	
		≤ 10	≤ 20
Universal Dependency Rules			
1	HDP-DEP	71.9	50.4
No Rules (Random Init)			
2	HDP-DEP	24.9	24.4
3	Headden et al. [44]	68.8	-
English-Specific Parsing Rules			
4	Deterministic (rules only)	70.0	62.6
5	HDP-DEP	**73.8**	**66.1**
[26] Rules			
6	Druck et al. [26]	61.3	-
7	HDP-DEP	64.9	42.2

Table 2.6: Directed accuracy of our model (HDP-DEP) on sentences of length 10 or less and 20 or less from WSJ with different rulesets and with no rules, along with various baselines from the literature. Entries in this table are numbered for ease of reference in the text.

threshold values from 75% to 90% yields results within 2.5% of each other, with a slight peak at 80%.

To further study the sensitivity of our method to how the threshold is set, we perform *post hoc* experiments with other threshold values on each of the other languages. As Figure 2-2 also shows, on average a value of 80% is optimal across languages, though again accuracy is stable within 2.5% between thresholds of 75% to 90%. These results demonstrate that a single threshold is broadly applicable across languages.

Interestingly, setting the threshold value independently for each language to its "true" proportion based on the gold dependencies (denoted as the "Gold" case in Figure 2-2) does not achieve optimal performance. Thus, knowledge of the true language-specific rule proportions is not necessary for high accuracy.

2.6.2 Analysis of Model Properties

We perform a set of additional experiments on English to gain further insight into HDP-DEP's behavior. Our choice of language is motivated by the fact that a wide range of prior parsing algorithms were developed for and tested exclusively on English. The experiments below demonstrate that 1) universal rules alone are powerful, but language- and dataset-tailored rules can further improve performance; 2) our model learns jointly from the rules and data, outperforming a rules-only deterministic parser; 3) the way we incorporate posterior constraints outperforms the generalized expectation constraint framework; and 4) our model exhibits low variance when seeded with different initializations. These results are summarized in Table 2.6 and discussed in detail below; line numbers refer to entries in Table 2.6. Each run of HDP-DEP below is with syntactic refinement enabled.

Impact of Rules Selection We compare the performance of HDP-DEP using the universal rules versus a set of rules designed for deterministically parsing the Penn Treebank (see Section 2.4 for details). As lines 1 and 5 of Table 2.6 show, language-specific rules yield better performance. For sentences of length 10 or less, the difference between the two rulesets is a relatively small 1.9%; for longer sentences, however, the difference is a substantially larger 15.7%. This is likely because longer sentences tend to be more complex and thus exhibit more language-idiosyncratic dependencies. Such dependencies can be better captured by the refined language-specific rules.

We also test model performance when no linguistic rules are available, i.e., performing unconstrained variational inference. The model performs substantially worse (line 2), confirming that syntactic category refinement in a fully unsupervised setup is challenging.

Learning Beyond Provided Rules Since HDP-DEP is provided with linguistic rules, a legitimate question is whether it improves upon what the rules encode, especially when the rules are complete and language-specific. We can answer this question by comparing the performance of our model seeded with the English-specific rules against a deterministic parser that implements the same rules. Lines 4 and 5 of Table 2.6 demonstrate that the model outperforms a rules-only deterministic parser by 3.8% for sentences of length 10 or less and by 3.5% for sentences of length 20 or less.

Comparison with Alternative Semi-supervised Parser The dependency parser based on the generalized expectation criteria [26] is the closest to our reported work in terms of technique. To compare the two, we run HDP-DEP using the 20 rules given by [26]. Our model achieves an accuracy of 64.9% (line 7) compared to 61.3% (line 6) reported in their work. Note that we do not rely on rule-specific expectation information as they do, instead requiring only a single expectation constraint parameter.[3]

Model Stability It is commonly acknowledged in the literature that unsupervised grammar induction methods exhibit sensitivity to initialization. As in the previous section, we find that the presence of linguistic rules greatly reduces this sensitivity: for HDP-DEP, the standard deviation over five randomly initialized runs with the English-specific rules is 1.5%, compared to 4.5% for the parser developed by [44] and 8.0% for DMV [47].

[3] As explained in Section 2.4, having a single expectation parameter is motivated by our focus on parsing with universal rules.

2.7 Conclusions and Subsequent Research

In this work we demonstrate that there is a universal aspect to dependency parsing that can be expressed in the form of high level syntactic universal rules. Moreover, syntactic universals encoded as declarative constraints improve grammar induction. Since the publication of this work in 2010 [60] a number of subsequent works have explored similar ideas.

A direct application of our universal rules is the heuristic based parsing system of Sogaard [74]. This system provides a deterministic method for applying the universal rules, the aim is to provide a more realistic baseline for parsers developed for low-resource languages. The results presented in the paper show that this simple baseline outperforms most unsupervised parsing systems submitted to the PASCAL challenge on grammar induction [36].

The idea of exploiting universal view of dependency grammar has also been explored further. Boonkwan and Steedman [11] proposed a set of universal prototypes that can be customized for a specific languages using high level knowledge of the language's syntax. The POS tags in these language specific prototypes are then manually mapped to language specific tags in target corpora. A limitation of this system, as well as our work, is that universal rules operate at the level of POS tags, thus requiring tag annotations for target language along with a mapping from language specific tags to universal tags.

One line of work has focused on developing universal POS tag mapping needed to enable the use of universal rules. Among these are the manually designed tagset mapping of Petrov et al. [65]. This work gives mapping for the tagsets of 25 treebanks from different languages. These mappings has been frequently used in parser transfer systems. Another such work is the mapping system of Zhang et al. [85]. This system automatically learns mappings from language specific to universal tagset, improving the performance of various transfer system compared to the case when manual mappings are used.

Chapter 3

Selective Sharing for Crosslingual Grammar Tranfer

Current top performing parsing algorithms rely on the availability of annotated data for learning the syntactic structure of a language. Standard approaches for extending these techniques to resource-lean languages either use parallel corpora or rely on annotated trees from other source languages. These techniques have been shown to work well for language families with many annotated resources (such as Indo-European languages). Unfortunately, for many languages there are no available parallel corpora or annotated resources in related languages. For such languages the only remaining option is to resort to unsupervised approaches, which are known to produce highly inaccurate results.

In this chapter, we present a new multilingual algorithm for dependency parsing. In contrast to previous approaches, this algorithm can learn dependency structures using annotations from a diverse set of source languages, even if this set is not related to the target language. In our *selective sharing* approach, the algorithm learns which aspects of the source languages are relevant for the target language and ties model parameters accordingly. This

approach is rooted in linguistic theory that characterizes the connection between languages at various levels of sharing. Some syntactic properties are universal across languages. For instance, nouns take adjectives and determiners as dependents, but not adverbs. However, the order of these dependents with respect to the parent is influenced by the typological features of each language.

To implement this intuition, we factorize generation of a dependency tree into two processes: selection of syntactic dependents and their ordering. The first component models the distribution of dependents for each part-of-speech tag, abstracting over their order. Being largely language-universal, this distribution can be learned in a supervised fashion from all the training languages. On the other hand, ordering of dependents varies greatly across languages and therefore should only be influenced by languages with similar properties. Furthermore, this similarity has to be expressed at the level of dependency types – i.e., two languages may share noun-adposition ordering, but differ in noun-determiner ordering. To systematically model this cross-lingual sharing, we rely on typological features that reflect ordering preferences of a given language. In addition to the known typological features, our parsing model embeds latent features that can capture cross-lingual structural similarities.

While the approach described so far supports a seamless transfer of shared information, it does not account for syntactic properties of the target language unseen in the training languages. For instance, in the CoNLL data, Arabic is the only language with the VSO ordering. To handle such cases, our approach augments cross-lingual sharing with unsupervised learning on the target languages.

We evaluated our selective sharing model on 17 languages from 10 language families. On this diverse set, our model consistently outperforms state-of-the-art multilingual dependency parsers. Performance gain, averaged over all the languages, is 5.9% when compared to the highest baseline. Our model achieves the most significant gains on non-Indo-

European languages, where we see a 14.4% improvement. We also demonstrate that in the absence of observed typological information, a set of automatically induced latent features can effectively work as a proxy for typology.

3.1 Related Work

Traditionally, parallel corpora have been a mainstay of multilingual parsing [82, 49, 69, 45, 83, 15, 71]. However, recent work in multilingual parsing has demonstrated the feasibility of transfer in the absence of parallel data. As a main source of guidance, these methods rely on the commonalities in dependency structure across languages. For instance, [60] explicitly encode these similarities in the form of universal rules which guide grammar induction in the target language. An alternative approach is to directly employ a non-lexicalized parser trained on one language to process a target language [84, 58, 73]. Since many unlexicalized dependencies are preserved across languages, these approaches are shown to be effective for related languages. For instance, when applied to the language pairs within the Indo-European family, such parsers outperform unsupervised monolingual techniques by a significant margin.

The challenge, however, is to enable dependency transfer for target languages that exhibit structural differences from source languages. In such cases, the extent of multilingual transfer is determined by the relation between source and target languages. [7] define such a relation in terms of phylogenetic trees, and use this distance to selectively tie the parameters of monolingual syntactic models. [19] do not use a predefined linguistic hierarchy of language relations, but instead learn the contribution of source languages to the training mixture based on the likelihood of the target language. [73] proposes a different measure of language relatedness based on perplexity between POS sequences of source

and target languages. Using this measure, he selects a subset of training source sentences that are closer to the target language. While all of the above techniques demonstrate gains from modeling language relatedness, they still underperform when the source and target languages are unrelated.

Our model differs from the above approaches in its emphasis on the selective information sharing driven by language relatedness. This is further combined with monolingual unsupervised learning. As our evaluation demonstrates, this layered approach broadens the advantages of multilingual learning to languages that exhibit significant differences from the languages in the training mix.

3.2 Linguistic Motivation

Language-Independent Dependency Properties Despite significant syntactic differences, human languages exhibit striking similarity in dependency patterns. For a given part-of-speech tag, the set of tags that can occur as its dependents is largely consistent across languages. For instance, adverbs and nouns are likely to be dependents of verbs, while adjectives are not. Thus, these patterns can be freely transferred across languages.

Shared Dependency Properties Unlike dependent selection, the ordering of dependents in a sentence differs greatly across languages. In fact, cross-lingual syntactic variations are primarily expressed in different ordering of dependents [42, 40]. Fortunately, the dimensions of these variations have been extensively studied in linguistics and are documented in the form of typological features [23, 43]. For instance, most languages are either dominantly prepositional like English or post-positional like Urdu. Moreover, a language may be close to different languages for different dependency types. For instance, Portuguese is a prepositional language like English, but the order of its noun-adjective dependency is

different from English and matches that of Arabic. Therefore, we seek a model that can express parameter sharing at the level of dependency types and can benefit from known language relations.

Language-specific Dependency Variations Not every aspect of syntactic structure is shared across languages. This is particularly true given a limited number of supervised source languages; it is quite likely that a target language will have previously unseen syntactic phenomena. In such a scenario, the raw text in the target language might be the only source of information about its unique aspects.

3.3 Model

We propose a probabilistic model for generating dependency trees that facilitates parameter sharing across languages. We assume a setup where dependency tree annotations are available for a set of source languages and we want to use these annotations to infer a parser for a target language. Syntactic trees for the target language are not available during training. We also assume that both source and target languages are annotated with a coarse parts-of-speech tagset which is shared across languages. Such tagsets are commonly used in multilingual parsing [84, 58, 73, 60].

The key feature of our model is a two-tier approach that separates the *selection* of dependents from their *ordering*:

1. *Selection Component:* Determines the dependent tags given the parent tag.
2. *Ordering Component:* Determines the position of each dependent tag with respect to its parent (right or left) and the order within the right and left dependents.

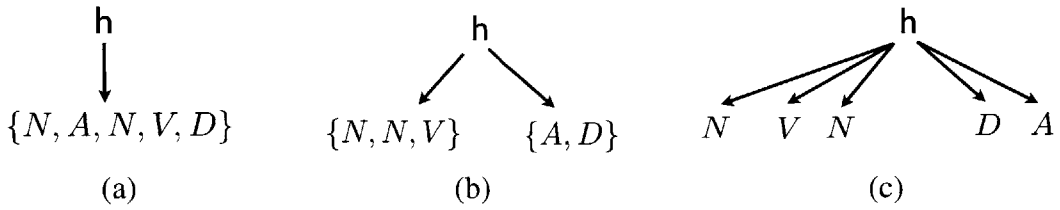


Figure 3-1: The steps of the generative process for a fragment with head h. In step (a), the unordered set of dependents is chosen. In step (b) they are partitioned into left and right unordered sets. Finally, each set is ordered in step (c).

This factorization constitutes a departure from traditional parsing models where these decisions are tightly coupled. By separating the two, the model is able to support different degrees of cross-lingual sharing on each level.

For the selection component, a reasonable approximation is to assume that it is the same for all languages. This is the approach we take here.

As mentioned in Section 3.2, the ordering of dependents is largely determined by the typological features of the language. We assume that we have a set of such features for every language l, and denote this feature vector by $\mathbf{v}_l$. We also experiment with a variant of our model where typological features are not observed. Instead, the model captures structural variations across languages by means of a small set of binary latent features. The values of these features are language dependent. We denote the set of latent features for language l by $\mathbf{b}_l$.

Finally, based on the well known fact that long distance dependencies are less likely [28], we bias our model towards short dependencies. This is done by imposing a corpus-level soft constraint on dependency lengths using the posterior regularization framework [38].

3.3.1 Generative Process

Our model generates dependency trees one fragment at a time. A *fragment* is defined as a subtree comprising the immediate dependents of any node in the tree. The process recursively generates fragments in a head outwards manner, where the distribution over fragments depends on the head tag. If the generated fragment is not empty then the process continues for each child tag in the fragment, drawing new fragments from the distribution associated with the tag. The process stops when there are no more non-empty fragments.

A fragment with head node h is generated in language l via the following stages:

- Generate the set of dependents of h via a distribution $P_{sel}(S|h)$. Here S is an unordered set of POS tags. Note that this part is universal (i.e., it does not depend on the language l).

- For each element in S decide whether it should go to the right or left of h as follows: for every $a \in S$, draw its direction from the distribution $P_{ord}(d|a, h, l)$, where $d \in \{R, L\}$. This results in two unordered sets S_R, S_L, the right and left dependents of h. This part *does* depend on the language l, since the relative ordering of dependents is not likely to be universal.

- Order the sets S_R, S_L. For simplicity, we assume that the order is drawn uniformly from all the possible unique permutations over S_R and S_L. We denote the number of such unique permutations of S_R by $n(S_R)$.[1] Thus the probability of each permutation of S_R is $\frac{1}{n(S_R)}$ [2].

[1]This number depends on the count of each distinct tag in S_R. For example if $S_R = \{N, N, N\}$ then $n(S_R) = 1$. If $S_R = \{N, D, V\}$ then $n(S_R) = 3!$.

[2]We acknowledge that assuming a uniform distribution over the permutations of the right and left dependents is linguistically counterintuitive. However, it simplifies the model by greatly reducing the number of parameters to learn.

Figure 3-1 illustrates the generative process. The first step constitutes the *selection* component and the last two steps constitute the *ordering* component. Given this generation scheme, the probability $P(D)$ of generating a given fragment D with head h will be:

$$P_{sel}(\{D\}|h) \prod_{a \in D} P_{ord}(d_D(a)|a, h, l) \frac{1}{n(D_R)n(D_L)} \tag{3.1}$$

Where we use the following notations:

- D_R, D_L denote the parts of the fragment that are to the left and right of h.
- $\{D\}$ is the unordered set of tags in D.
- $d_D(a)$ is the position (either R or L) of the dependent a w.r.t. the head of D.

In what follows we discuss the parameterizations of the different distributions.

Selection Component The selection component draws an unordered set of tags S given the head tag h. We assume that the process is carried out in two steps. First the number of dependents n is drawn from a distribution:

$$P_{size}(n|h) = \theta_{size}(n|h) \tag{3.2}$$

where $\theta_{size}(n|h)$ is a parameter for each value of n and h. We restrict the maximum value of n to four, since this is a reasonable bound on the total number of dependents for a single parent node in a tree. These parameters are non-negative and satisfy $\sum_n \theta_{size}(n|h) = 1$. In other words, the size is drawn from a categorical distribution that is fully parameterized.

Next, given the size n, a set S with $|S| = n$ is drawn according to the following log-linear

model:

$$P_{set}(S|h,n) = \frac{1}{Z_{set}(h,n)} e^{\sum_{S_i \in S} \theta_{sel}(S_i|h)}$$
$$Z_{set}(h,n) = \sum_{S:|S|=n} e^{\sum_{S_i \in S} \theta_{sel}(S_i|h)}$$

In the above, S_i is the i^{th} POS tag in the unordered set S, and $\theta_{sel}(S_i|h)$ are parameters. Thus, large values of $\theta_{sel}(S_i|h)$ indicate that POS S_i is more likely to appear in the subset with parent POS h.

Combining the above two steps we have the following distribution for selecting a set S of size n:

$$P_{sel}(S|h) = P_{size}(n|h) P_{set}(S|h,n) \; . \tag{3.3}$$

ID	Feature Description	Values
81A	Order of Subject, Object and Verb	SVO, SOV, VSO, VOS, OVS, OSV
85A	Order of Adposition and Noun	Postpositions, Prepositions, Inpositions
86A	Order of Genitive and Noun	Genitive-Noun, Noun-Genitive
87A	Order of Adjective and Noun	Adjective-Noun, Noun-Adjective
88A	Order of Demonstrative and Noun	Demonstrative-Noun, Noun-Demonstrative

Table 3.1: The set of typological features that we use in our model. For each feature, the first column gives the ID of the feature as used in WALS, the second column describes the feature and the last column enumerates the allowable values for the feature. Besides these values, each feature can also have a value of 'No dominant order'.

Ordering Component The ordering component consists of distributions $P_{ord}(d|a,h,l)$ that determine whether tag a will be mapped to the left or right of the head tag h. We model

it using the following log-linear model:

$$P_{ord}(d|a,h,l) = \frac{1}{Z_{ord}(a,h,l)} e^{\mathbf{w}_{ord} \cdot \mathbf{g}(d,a,h,\mathbf{v}_l)}$$
$$Z_{ord}(a,h,l) = \sum_{d \in \{R,L\}} e^{\mathbf{w}_{ord} \cdot \mathbf{g}(d,a,h,\mathbf{v}_l)}$$

Note that in the above equations the ordering component depends on the known typological features $\mathbf{v}_l$. In the setup when typological features are not known, $\mathbf{v}_l$ is replaced with the latent ordering feature set $\mathbf{b}_l$.

The feature vector $\mathbf{g}$ contains indicator features for combinations of a, h, d and individual features v_{li} (i.e., the i^{th} typological features for language l).

3.3.2 Typological Features

The typological features we use are a subset of order-related typological features from "The World Atlas of Language Structure" [43]. We include only those features whose values are available for all the languages in our dataset. Table 3.1 summarizes the set of features that we use. Note that we do not explicitly specify the correspondence between these features and the model parameters. Instead, we leave it for the model to learn this correspondence automatically.

3.3.3 Dependency Length Constraint

To incorporate the intuition that long distance dependencies are less likely, we impose a posterior constraint on dependency length. In particular, we use the Posterior Regularization (PR) framework of [38]. The PR framework incorporates constraints by adding a

penalty term to the standard likelihood objective. This term penalizes the distance of the model posterior from a set $\mathcal{Q}$, where $\mathcal{Q}$ contains all the posterior distributions that satisfy the constraints. In our case the constraint is that the expected dependency length is less than or equal to a pre-specified threshold value b. If we denote the latent dependency trees by z and the observed sentences by x then

$$\mathcal{Q} = \{q(z|x) : E_q[f(x,z)] \leq b\} \tag{3.4}$$

where $f(x,z)$ computes the sum of the lengths of all dependencies in z with respect to the linear order of x. We measure the length of a dependency relation by counting the number of tokens between the head and its modifier. The PR objective penalizes the KL-divergence of the model posterior from the set $\mathcal{Q}$:

$$\mathcal{L}_\theta(x) - \mathrm{KL}\left(\mathcal{Q} \parallel p_\theta(z|x)\right)$$

where θ denotes the model parameters and the first term is the log-likelihood of the data. This objective can be optimized using a modified version of the EM algorithm [38].

3.4 Parameter Learning

Our model is parameterized by the parameters θ_{sel}, θ_{size} and $\mathbf{w}_{ord}$. We learn these by maximizing the likelihood of the training data. As is standard, we add ℓ_2 regularization on the parameters and tune it on source languages. The likelihood is marginalized over all latent variables. These are:

- For sentences in the target language: all possible derivations that result in the observed POS tag sequences. The derivations include the choice of unordered sets

size n, the unordered sets themselves S, their left/right allocations and the orderings within the left and right branches.

- For all languages: all possible values of the latent features $\mathbf{b}_l$.[3]

Since we are learning with latent variables, we use the EM algorithm to monotonically improve the likelihood. At each E step, the posterior over latent variables is calculated using the current model. At the M step this posterior is used to maximize the likelihood over the fully observed data. To compensate for the differences in the amount of training data, the counts from each language are normalized before computing the likelihood.

The M step involves finding maximum likelihood parameters for log-linear models in Equations 3.3 and 3.4. This is done via standard gradient based search; in particular, we use the method of BFGS.

We now briefly discuss how to calculate the posterior probabilities. For estimating the $\mathbf{w}_{ord}$ parameters we require marginals of the type $P(b_{li}|\mathcal{D}_l; \mathbf{w}^t)$ where $\mathcal{D}_l$ are the sentences in language l, b_{li} is the i_{th} latent feature for the language l and $\mathbf{w}^t$ are the parameter values at iteration t. Consider doing this for a source language l. Since the parses are known, we only need to marginalize over the other latent features. This can be done in a straightforward manner by using our probabilistic model. The complexity is exponential in the number of latent features, since we need to marginalize over all features other than b_{li}. This is feasible in our case, since we use a relatively small number of such features.

When performing unsupervised learning for the target language, we need to marginalize over possible derivations. Specifically, for the M step, we need probabilities of the form $P(\text{a modifies h}|\mathcal{D}_l; \mathbf{w}^t)$. These can be calculated using a variant of the inside outside algorithm (see Appendix A.1). The exact version of this algorithm would be exponential in

[3]This corresponds to the case when typological features are not known.

the number of dependents due to the $\frac{1}{n(S_r)}$ term in the permutation factor. Although it is possible to run this exact algorithm in our case, where the number of dependents is limited to 4, we use an approximation that works well in practice: instead of $\frac{1}{n(S_r)}$ we use $\frac{1}{|S_r|!}$. In this case the runtime is no longer exponential in the number of children, so inference is much faster.

Finally, given the trained parameters we generate parses in the target language by calculating the maximum a posteriori derivation. This is done using a variant of the CKY algorithm.

3.5 Experimental Setup

Datasets and Evaluation We test the effectiveness of our approach on 17 languages: Arabic, Basque, Bulgarian, Catalan, Chinese, Czech, Dutch, English, German, Greek, Hungarian, Italian, Japanese, Portuguese, Spanish, Swedish and Turkish. We used datasets distributed for the 2006 and 2007 CoNLL Shared Tasks [14, 62]. Each dataset provides manually annotated dependency trees and POS tags. To enable crosslingual sharing, we map the gold part-of-speech tags in each corpus to a common coarse tagset [84, 73, 58, 60]. The coarse tagset consists of 11 tags: noun, verb, adjective, adverb, pronoun, determiner, adposition, numeral, conjunction, particle, punctuation mark, and X (a catch-all tag). Among several available fine-to-coarse mapping schemes, we employ the one of [60] that yields consistently better performance for our method and the baselines than the mapping proposed by [65]. In most of our experiments, we assume the availability of typological feature values for each language (Table 3.2).

As the evaluation metric, we use directed dependency accuracy. Following standard evaluation practices, we do not evaluate on punctuation. For both the baselines and our model

ID	Name	ar	ba	bu	ca	ch	cz	du	en	ge	gr	hu	it	ja	po	sp	sw	tu
88A	Dem-Noun	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
88A	Noun-Dem	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
87A	Adj-Noun	0	0	1	0	1	1	1	1	1	1	1	0	1	0	0	1	1
87A	Noun-Adj	1	1	0	1	0	0	0	0	0	0	0	1	0	1	1	0	0
85A	No order	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
85A	Post	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1
85A	Pre	1	0	1	1	0	1	1	1	1	1	0	1	0	1	1	1	0
86A	Gen-Noun	0	1	0	0	1	0	0	0	0	0	1	0	1	0	0	1	1
86A	No order	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0
86A	Noun-Gen	1	0	0	1	0	0	1	0	1	1	0	1	0	1	1	0	0
81A	SOV	0	1	0	0	0	0	1	0	1	0	1	0	1	0	0	0	1
81A	SVO	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0
81A	VSO	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Table 3.2: Typological feature values for Arabic (ar), Basque (ba), Bulgarian (bu), Catalan (ca), Chinese (ch), Czech (cz), Dutch (du), English (en), German (ge), Greek (gr), Hungarian (hu), Italian (it), Japanese (ja), Portuguese (po), Spanish (sp), Swedish (sw) and Turkish (tu). Dem = Demonstrative, Adj = Adjective, Pre = Preposition, Post = Postposition, Gen = Genitive, S = Subject, O = Object, V = Verb and "No order" = No dominant order

we evaluate on all sentences of length 50 or less ignoring punctuation.

Training Regime Our model typically converges quickly and does not require more than 50 iterations of EM. When the model involves latent typological variables, the initialization of these variables can impact the final performance. As a selection criterion for initialization, we consider the performance of the final model averaged over the supervised source languages. We perform ten random restarts and select the best according to this criterion. Likewise, the threshold value b for the PR constraint on the dependency length is tuned on the source languages, using average test set accuracy as the selection criterion.

Baselines We compare against the state-of-the-art multilingual dependency parsers that do not use parallel corpora for training. All the systems were evaluated using the same fine-to-coarse tagset mapping. The first baseline, *Transfer*, uses direct transfer of a dis-

criminative parser trained on all the source languages [58]. This simple baseline achieves surprisingly good results, within less than 3% difference from a parser trained using parallel data. In the second baseline (*Mixture*), parameters of the target language are estimated as a weighted mixture of the parameters learned from annotated source languages [19]. The underlying parsing model is the dependency model with valance (DMV) [47]. Originally, the baseline methods were evaluated on different sets of languages using a different tag mapping. Therefore, we obtained new results for these methods in our setup. For the *Transfer* baseline, for each target language we trained the model on all other languages in our dataset. For the *Mixture* baseline, we trained the model on the same four languages used in the original paper — English, German, Czech and Italian. When measuring the performance on these languages, we selected another set of four languages with a similar level of diversity.[4]

3.6 Results

Table 3.3 summarizes the performance for different configurations of our model and the baselines.

3.6.1 Comparison against Baselines

On average, the selective sharing model outperforms both baselines, yielding 8.9% gain over the weighted mixture model [19] and 5.9% gain over the direct transfer method [58]. Our model outperforms the weighted mixture model on 15 of the 17 languages and the transfer method on 12 of the 17 languages. Most of the gains are obtained on non-Indo-

[4]We also experimented with a version of the [19] model trained on all the source languages. This setup resulted in decreased performance. For this reason, we chose to train the model on the four languages.

European languages, that have little similarity with the source languages. For this set, the average gain over the transfer baseline is 14.4%. With some languages, such as Japanese, achieving gains of as much as 30%.

On Indo-European languages, the model performance is almost equivalent to that of the best performing baseline. To explain this result we consider the performance of the supervised version of our model which constitutes an upper bound on the performance. The average accuracy of our supervised model on these languages is 66.8%, compared to the 76.3% of the unlexicalized MST parser. Since Indo-European languages are overrepresented in our dataset, a target language from this family is likely to exhibit more similarity to the training data. When such similarity is substantial, the transfer baseline will benefit from the power of a context-rich discriminative parser.

A similar trait can be seen by comparing the performance of our model to an oracle version of our model which selects the optimal source language for a given target language (column 7). Overall, our method performs similarly to this oracle variant. However, the gain for non Indo-European languages is 1.9% vs -1.3% for Indo-European languages.

3.6.2 Analysis of Model Properties

We first test our hypothesis about the universal nature of the dependent selection. We compare the performance of our model (column 6) against a variant (column 8) where this component is trained from annotations on the target language. The performance of the two is very close – 1.8%, supporting the above hypothesis.

To assess the contribution of other layers of selective sharing, we first explore the role of typological features in learning the ordering component. When the model does not have access to observed typological features, and does not use latent ones (column 4),

the accuracy drops by 2.6%[5]. For some languages (e.g., Turkish) the decrease is very pronounced. Latent typological features (column 5) do not yield the same gain as observed ones, but they do improve the performance of the typology-free model by 1.4%.

Next, we show the importance of using raw target language data in training the model. When the model has to make all the ordering decisions based on meta-linguistic features without account for unique properties of the target languages, the performance decreases by 0.9% (see column 3).

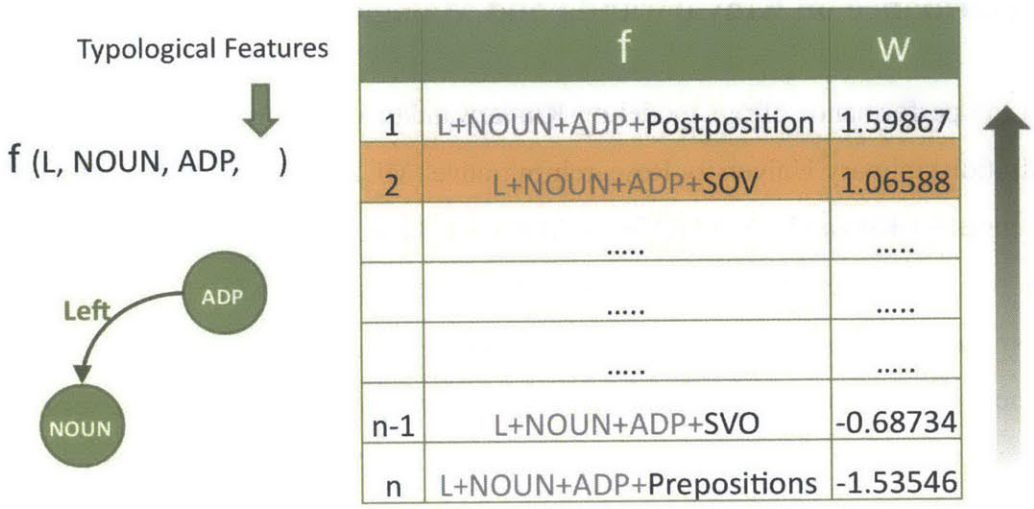
To assess the relative difficulty of learning the ordering and selection components, we consider model variants where each of these components is trained using annotations in the target language. As shown in columns 8 and 9, these two variants outperform the original model, achieving 61.3% for supervised selection and 63.7% for supervised ordering. Comparing these numbers to the accuracy of the original model (column 6) demonstrates the difficulty inherent in learning the ordering information. This finding is expected given that ordering involves selective sharing from multiple languages.

Overall, the performance gap between the selective sharing model and its monolingual supervised counterpart is 7.3%. In contrast, the unsupervised monolingual variant of our model achieves a meager 26%.[6] This demonstrates that our model can effectively learn relevant aspects of syntactic structure from a diverse set of languages.

3.6.3 Analysis of Typological Feature Weights

We analyzed the learned feature weights for the feature involving typological properties. We found that for every dependency type, highest feature weight is learned when it is combined with its corresponding typological property. For instance, Figure 3.6.3 shows

[5]In this setup, the ordering component is trained in an unsupervised fashion on the target language.
[6]This performance is comparable to other generative models such as DMV [47].



	f	W
1	L+NOUN+ADP+Postposition	1.59867
2	L+NOUN+ADP+SOV	1.06588
	.....	.....
	.....	.....
	.....	.....
n-1	L+NOUN+ADP+SVO	-0.68734
n	L+NOUN+ADP+Prepositions	-1.53546

Figure 3-2: Typological feature weights learned by the model for postpositional dependencies between Nouns and Adpositions

weights for features corresponding to the dependency between Adpositions and Nouns when Noun is to the left of Adposition. Model learns correctly to assign highest weight to the feature that combines this dependency with Postposition property, and the lowest weight when combined with the Preposition property. Furthermore, the second highest weight is learned for the feature that combines this dependency with SVO property. This corresponds to the known language universal that SVO languages are always postpositional. Although the experiments presented here are not designed to unveil language universals, this analysis suggests that linguistically motivated models can be used to uncover correlations between different inter and intra-language phenomena.

3.6.4 Performance on Kinyarwanda and Malagasy

We also test the performance of our model on Kinyarwanda and Malagasy data produced as part of Multidisciplinary University Research Initiative (MURI)[7]. This data has 270 annotated sentences for Kinyarwanda and 156 for Malagasy. In these experiments, all the languages from the previous experiments were used as source languages while Kinyarwanda and Malagasy each were used as target language. The results are shown in Figure 3.6.4. The plot at the top shows the results when gold part-of-speech tags are available, the results on the bottom are produced using automatically annotated POS tags [35]. In both cases, our model outperforms the parser transferred directly from the closest source language by 4% and 10% respectively for Malagasy and Kinyarwanda. Moreover, the performance of selective-sharing model is quite close to its supervised counterpart.

3.7 Conclusions and Subsequent Research

We present a novel algorithm for multilingual dependency parsing that uses annotations from a diverse set of source languages to parse a new unannotated language. Overall, our model consistently outperforms the multi-source transfer based dependency parser of [58]. Our experiments demonstrate that the model is particularly effective in processing languages that exhibit significant differences from the training languages.

One limitation of this work is the loss of language specific information that is available to supervised parsers in the form of language specific tags and words. Täckström et al. [80] propose a method for incorporating typological features along with language family information into a discriminative model. This method improves over our work achieving a performance of 62%. The performance is further improved by using the target language

[7]http://www.linguisticcore.info

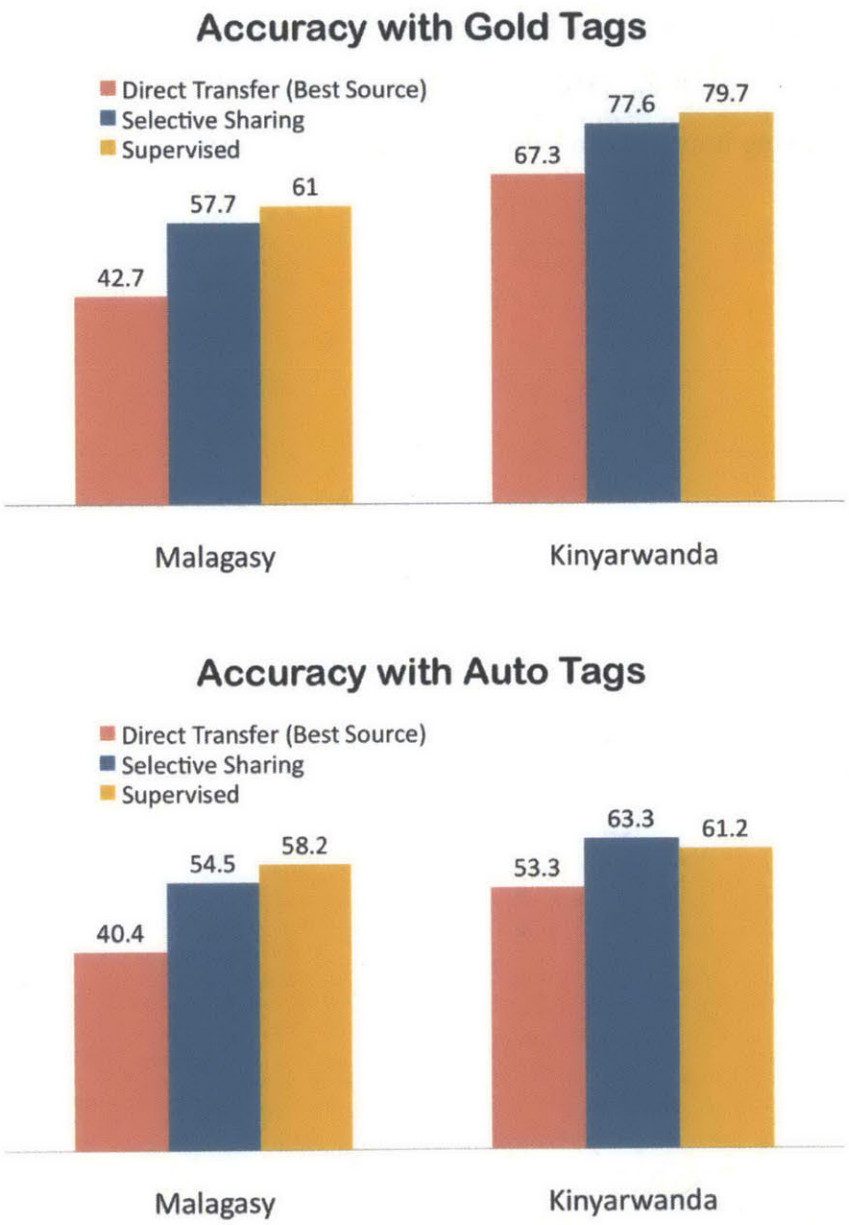


Figure 3-3: Performance of our Selective Sharing model (middle column) compared against Direct Transfer from the best source (left) and Supervised (right) using gold POS tags (top) and automatically generated POS tags (bottom)

data to adapt the transfer parser achieving an accuracy of 65%. The gain in performance incurred by incorporating language specific unannotated data suggests that crosslingual mapping of lexical dependencies may further improve the quality of transfer. A work parallel to ours by Täckström et al. [80] suggest one such method of learning crosslingual lexical clusters, showing improvements over previous direct transfer parsers on a subset of CoNLL data.

	Baselines		Selective Sharing Model							
	Mixture	Transfer	(D-,T_o)	(D+)	(D+,T_l)	(D+,T_o)	Best Pair	Sup. Sel.	Sup. Ord.	MLE
Catalan	64.9	69.5	71.9	66.1	66.7	71.8	74.8	70.2	73.2	72.1
Italian	61.9	68.3	68.0	65.5	64.2	65.6	68.3	65.1	70.7	72.3
Portuguese	72.9	75.8	76.2	72.3	76.0	73.5	76.4	77.4	77.6	79.6
Spanish	57.2	65.9	62.3	58.5	59.4	62.1	63.4	61.5	62.6	65.3
Dutch	50.1	53.9	56.2	56.1	55.8	55.9	57.8	56.3	58.6	58.0
English	45.9	47.0	47.6	48.5	48.1	48.6	44.4	46.3	60.0	62.7
German	54.5	56.4	54.0	53.5	54.3	53.7	54.8	52.4	56.2	58.0
Swedish	56.4	63.6	52.0	61.4	60.6	61.5	63.5	67.9	67.1	73.0
Bulgarian	67.7	64.0	67.6	63.5	63.9	66.8	66.1	66.2	69.5	71.0
Czech	39.6	40.3	43.9	44.7	45.4	44.6	47.5	53.2	51.2	58.9
Arabic	44.8	40.7	57.2	58.8	60.3	58.9	57.6	62.9	61.9	64.2
Basque	32.8	32.4	39.7	40.1	39.8	47.6	42.0	46.2	47.9	51.6
Chinese	46.7	49.3	59.9	52.2	52.0	51.2	65.4	62.3	65.5	73.5
Greek	56.8	60.4	61.9	67.5	67.3	67.4	60.6	67.2	69.0	70.5
Hungarian	46.8	54.3	56.9	58.4	58.8	58.5	57.0	57.4	62.0	61.6
Japanese	33.5	34.7	62.3	56.8	61.4	64.0	54.8	63.4	69.7	75.6
Turkish	28.3	34.3	59.1	43.6	57.8	59.2	56.9	66.6	59.5	67.6
Average	50.6	53.6	58.6	56.9	58.3	**59.5**	59.5	61.3	63.7	66.8

Table 3.3: Directed dependency accuracy of different variants of our selective sharing model and the baselines. The first section of the table (column 1 and 2) shows the accuracy of the weighted mixture baseline [19] (Mixture) and the multi-source transfer baseline [58] (Transfer). The middle section shows the performance of our model in different settings. D$\pm$ indicates the presence/absence of raw target language data during training. T_o indicates the use of observed typological features for all languages and T_l indicates the use of latent typological features for all languages. The last section shows results of our model with different levels of oracle supervision: a. (Best Pair) Model parameters are borrowed from the best source language based on the accuracy on the target language b. (Sup. Sel.) Selection component is trained using MLE estimates from target language c. (Sup. Ord.) Ordering component is trained using MLE estimates from the target language d. (MLE) All model parameters are trained on the target language in a supervised fashion. The horizontal partitions separate language families. The first three families are sub-divisions of the Indo-European language family.

Chapter 4

Many Views in One: Dependency Parsing Using Corpus Level Models

Combining multiple ways of solving the same task rarely fails. This is particularly true when models are simple, misspecified, and capture largely complementary aspects of the data. The insight has been used frequently in unsupervised parsing. For example, early successes [47] sought to combine dependency and constituent parsing. Another perspective to combining models is to incorporate declarative knowledge, insufficient on its own, into an otherwise unsupervised model [21, 26, 60].

There are many technical ways to realize this basic intuition. For example, [47] used products of experts where each expert parser was based on a single grammar formalisms. The product form primarily reinforces shared predictions while enabling each model to dominate the other, ignorant one. Declarative knowledge can be incorporated via posterior regularization [60] or generalized expectation [26]. In either case, the unsupervised model is guided towards predefined statistical properties.

Multiple views, types of models or declarative knowledge, should be possible to incorporate directly into a single model where their strengths and weaknesses would be balanced in the same context. Such an approach should be possible to formulate in a Bayesian framework where the full range of parameters (and their combinations) are explored. It is less obvious, however, whether such an approach remains computationally feasible.

We propose here unsupervised Bayesian models of dependency parsing that operate on the corpus level. By dispensing with the per-sentence view of modeling, we can easily merge multiple ways of scoring or constraining parsing decisions. We begin with a simple generative dependency parsing model, akin to [47], and adorn it with various complementary views of scoring. Different views are incorporated as additional parameters that pertain to the same parsing decisions. By integrating over the parameters (i.e., exploring jointly optimal parameter settings), we necessarily couple parsing decisions across the sentences in the corpus. It is still possible, however, to sample a new improved parse for each sentence, one sentence at a time, in a (collapsed) Gibbs' sampling framework, or approximately using Metropolis-Hastings. We experiment with several such alternative views.

We evaluate our method on 19 languages, using dependency data from CONLL 2007 and CONLL 2006 datasets. For unsupervised experiments, we compare our results against the state-of-the-art parser [76] which combines several parsing models. On average, our model outperforms this baseline by 1.9%. However, their relative performance across individual languages varies greatly, suggesting that the two parsers have different modeling strengths. In addition, we show that our model outperforms posterior regularization applied with the same set of constraints.

We also evaluate our model in a semi-supervised set-up, providing it with a small amount of training data. Across different sizes of training data, our model shows consistent improvement over purely supervised system trained on the same amount of annotated data

but without any unlabeled text. For instance, given 10 training sentences augmented with raw data, the semi-supervised model achieves 67.2%, compared to 63.2% of the fully supervised model.

4.1 Related Work

Models based on multiple views have been extensively studied in supervised parsing [29, 64, 39], and more broadly in machine learning [9, 24, 18]. In this section, we focus on how this approach has been explored in unsupervised generative parsers. We first discuss the types of overlapping linguistic constraints that were successfully captured in this type of models. Then we discuss mechanisms for effectively leveraging these overlapping views.

Linguistic Information Encoded in Overlapping Views A natural way to leverage overlapping views is to combine representations from different grammar formalisms into a single model. An early instance of this approach is a work by [47] that learns a model that combines constituency and dependency components. While the trees produced by these formalisms are different, they can constrain each other: the boundaries of constituents should be aligned with projected spans of the nodes in dependency trees. The desired structure should be preferred by both formalisms. The same ideas have been implemented in other models [75, 59].

In the above case, overlapping views are jointly learned from data. In other approaches, one of these overlapping representations has been specified declaratively, rather than learned. Such representations may specify high-level linguistic constraints on dependency relations. Examples include the types of modifiers a head verb can take. This type of rules have been successfully used to constrain parsing models learned from raw data [21, 60, 12, 26, 67].

Since these rules often include expected frequencies, the model is able to incorporate them as soft constraints.

Mechanisms for Combining Overlapping Views One common mechanism for handling multiple views in a generative framework is the product of experts [47, 72, 13]. In this scenario, each view forms a generative model and the latent structure is scored using the product of their probabilities. These models are simple and effective in practice. However, product of experts models typically require some form of approximation. For instance, [47] and [72] approximate the inference by assuming that each model independently generates the data; this results in a deficient model. [13], on the other hand, approximate the inference using a fully-factorized variational distribution, thus optimizing a lower bound on true objective.

Another approach for combining overlapping views is to use posterior regularization [34] and generalized expectation criteria [26]. These techniques are used when one of the views is observed, and specified as a constraint with an expected value. These constraints may express patterns that involve non-local structures that are hard to capture by a local generative model. This framework keeps the generative model intact, using the constraints to limit the search space during learning. This modular architecture can encourage the constraints to a moderate degree: if the basic model cannot satisfy the constraints, posterior regularization cannot move it in the right direction. In contrast, our approach incorporates different views in a single model, and thereby has means to find a solution consistent with all the views.

Yet another way of incorporating overlapping views is to utilize locally normalized log-linear models of Berg-Kirkpatrick et al. [6]. While these log-linear models allow multiple views via overlapping features, they can do so only in a local context. Adding features pertaining to bigger contexts (sentence-level or corpus-level) will introduce an intractable normalization term. However, our approach allows the use of non-local parameters without

making inference intractable.

More recently, [76] proposed an effective way for incorporating multiple views via model combination. Different views are represented in separate models, which are then arranged into a network via a series of model perturbations and output-merge operations. Each subsequent parser in the pipeline benefits from the preceding simpler parsers. Our approach, on the other hand, learns all the views jointly, thereby allowing each view to inform the others.

4.2 Basic Dependency Model

Our model builds on a simple base model. As the base model, we use an extended Bayesian version of the generative dependency model introduced in [47]. Schematically, the generative process starts at the dedicated root node. The modifier nodes of each node in the tree are generated recursively in a head-outward and depth-first manner. The generation of the modifiers for a node involves two types of decisions. First, at each step, a decision is made whether to stop or continue generating modifiers (in either direction). Next, in case of a continue decision, each modifier is generated from the set of possible modifiers. The resulting tree is projective.

Model parameters The binary stop/continue decision for generating modifiers is conditioned on other available features in the tree so far. The probability that we stop generating modifiers is specified by a multi-way parameter table $\theta_{stop}(h, dir, args, dist) \in [0, 1]$. Here h is the part-of-speech tag of the head node; dir refers to the direction of generation (right or left) with respect to the head; $args$ is the number of arguments generated so far on the dir side of the head node; finally, $dist$ is the bucketed distance from the head to the outer most leaf node generated so far on side dir. Note that the number of arguments, variable

$args$, includes only modifiers that are indeed arguments (as opposed to adjuncts). This is made possible by means of a small set of rules (see Table 4.1) used to distinguish argument modifiers from all other modifiers [1]. $dist$ is another type of information that is not used in standard generative dependency models, including [47]. However, as noted by [75], a depth-first generative process allows to condition on any information pertaining to the subtrees generated by the preceding sibling nodes on the same side.

Once the decision is made to generate another modifier node, its part-of-speech tag is generated from a multinomial distribution $\theta_{choose}(t|h, dir)$ where t ranges over the possible tags. Note that we do not have a separate parameter governing the direction of generation. Indeed, this information is folded in stop/continue decisions. To complete the generative process, we can simply determine, as a default, that one side (e.g., right) is generated first.

Our model is Bayesian and therefore requires prior distributions over the parameters. Both θ_{stop} and θ_{choose} distributions have symmetric Dirichlet priors with parameters θ^0_{stop} and θ^0_{choose}, respectively. The prior distributions are assumed to be independent of each other for each setting of the conditioning variables. Thus, for example, the prior distribution over the multinomial distribution $\theta_{choose}(\cdot|h, dir)$ is distinct from $\theta_{choose}(\cdot|h', dir)$, whenever $h \neq h'$.

Model over the corpus We denote an annotated corpus as $\{(x_i, y_i)\}_{i=1}^N$, where x_i is a sentence and y_i the corresponding parse. In this paper, our focus is on unsupervised (and semi-supervised) parsing. We are therefore primarily given only the sentences $\{x_i\}_{i=1}^N$. In this case, y_i represents a predicted parse for x_i.

According to our base model described above, each sentence x_i would be represented as a sequence of tags. In this case, we can evaluate the probability $P(y_i, x_i|\theta)$ corresponding to

[1] These rules are only used to decide the type of a modifier given both the head and the modifier. They do not help select the modifiers for a head.

Head	Argument
Verb	Noun
Verb	Pronoun
Verb	Adjective
Verb	Verb
Adposition	Noun
Noun	Determiner

Table 4.1: The set of rules used to distinguish between arguments and adjuncts. Only modifiers that correspond to one of the head-modifier pairs listed in the table are considered arguments; all other modifiers are assumed to be adjuncts.

each valid parse $y_i \in \mathcal{Y}(x_i)$. θ here refers to the set of all model parameters. The Bayesian parsing model therefore induces a distribution over the corpus as a whole

$$P(y_1, \ldots, y_N, x_1, \ldots, x_N) = \int P(\theta) \left[\prod_{i=1}^{N} P(y_i, x_i|\theta) \right] d\theta \tag{4.1}$$

Our goal is to find likely realizations of $\{y_i\}_{i=1}^N$ given the sentences $\{x_i\}_{i=1}^N$, i.e., to maximize the joint distribution $P(y_1, \ldots, y_N, x_1, \ldots, x_N)$ with respect to $\{y_i\}_{i=1}^N$. We will next extend the corpus level model.

4.3 Modeling Overlapping Decisions

The key idea behind our approach is to incorporate additional parameters that are used to guide overlapping decisions in the basic dependency model. For example, we will introduce parameters for generating a coarse part-of-speech tag for each modifier as well as its

refined tag. The two decisions are clearly related but can be modeled in terms of two separate sets of parameters, and independent prior distributions over those parameters. Despite apparent over-generation of decisions, we can define and use effectively a consistent joint distribution on the corpus level.

Consider a simple illustrative example. Let t index refined part-of-speech tags and t_c coarse tags. The two indexes are logically tied such that $C(t) = t_c$ for any t. Consider a sequence of refined tags $t_1, \ldots, t_N$. We specify a joint distribution over these tags in an overlapping manner. Specifically, we can define $P(t_1, \ldots, t_N)$ as

$$\frac{1}{Z} \int P(\theta) P_c(\theta_c) \left[\prod_{i=1}^{N} \theta(t_i) \theta_c(C(t_i)) \right] d\theta d\theta_c \tag{4.2}$$

$$= \frac{1}{Z} \left[\int P(\theta) \prod_{i=1}^{N} \theta(t_i) d\theta \right] \left[\int P_c(\theta_c) \prod_{i=1}^{N} \theta_c(C(t_i)) d\theta \right] \tag{4.3}$$

where θ are the multinomial parameters over the refined tags and θ_c the corresponding parameters over the coarse tags. By defining a globally normalized joint distribution, we essentially draw all data nodes and parameter nodes at once from this distribution, hence no over-generation. This formulation can be viewed as a factor graph where a single factor connects all nodes. The potential associated with this factor is defined in terms of multinomial distributions and their priors.

Note that the feasibility of carrying out the integrals in 4.3 is independent of how many overlapping parameters we have. If the prior distributions are conjugate priors, the integrals can be carried out in closed form. The overall normalization constant Z is never needed if the model is used in a Gibbs' sampling framework where we would sample one tag t_i given the others $t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_N$. We extend this basic approach for parsing with overlapping parameters.

4.4 The Full model

Our full model is comprised of the basic dependency model and a set of additional overlapping parameters. These additional parameters pertain to different aspects of dependency trees that overlap with the basic model and with each other.

In a semi-supervised setting, multiple overlapping views of latent structure are helpful since each view serves as a regularizer (sanity check) for the others. For the unsupervised setup, this approach allows us to incorporate high level declarative knowledge about dependency patterns.

The next sections discuss the aspects of dependency structures that we wish to capture using additional overlapping views of the data. We also discuss the linguistic intuitions that motivated our choices.

4.4.1 Dependency Parameters

We augment the basic dependency model with lexical and coarse-tag versions of θ_{stop} and θ_{choose}. The parameters and their prior distributions are otherwise defined analogously to the base model.

4.4.2 Constituency Parameters

The constituency grammar formalism is based on the idea of substitutability, i.e. constituents of the same type can be substituted for each other to form syntactically valid sentences. In this work, we integrate a notion of constituency into our model. In particular, we introduce parameters that can capture contextual patterns seen in the constituents

induced by dependency trees. Thus capitalizing on the idea of substitutability that forms the basis of the constituency formalism.

For a given sentence, a projective dependency tree also gives partial constituent bracketing. The projected span of each node in the dependency tree corresponds to one of the brackets in the corresponding constituency tree. This means we can expect these brackets to exhibit the same substitutability behavior.

To incorporate this intuition, we introduce two parameters. The first parameter looks at the boundary tags of each constituent span. In particular, for every coarse part-of-speech tag t with coarse head tag h we have a multinomial distribution $\phi_{const}(t_{B,L}, t_{B,R}|h, t)$ over pairs of coarse tags. Where $t_{B,L}$ and $t_{B,R}$ are the left and right boundary tags of the constituent span projected by the dependency sub-tree rooted at t. In other words, this distribution characterizes a sub-tree for the pair (h, t) based solely on its boundary leaf tags.

The second parameter captures the context of a constituency span. Again, for every parent-tag pair (h, t) we have a multinomial distribution $\phi_{context}(t_{C,L}, t_{C,R}|h, t)$ over the pairs of coarse tags that occur outside of the projected span on both sides. Both ϕ_{const} and $\phi_{context}$ distributions have symmetric Dirichlet priors with parameters ϕ^0_{const} and $\phi^0_{context}$. Figure 4-1 illustrates these parameters.

4.4.3 Rule-Based Distribution

In rule-based distribution, we reuse the idea of universal linguistic patterns from Chapter 2. However, the way these rules are incorporated is different from our previous approach. Instead of using rules as constraints during inference, we add them as a model parameter.

We model the ratio of dependencies in a sentence that match one of the pre-specified coarse rules (see Table 4.2). We assume that this ratio follows a Gaussian distribution with mean

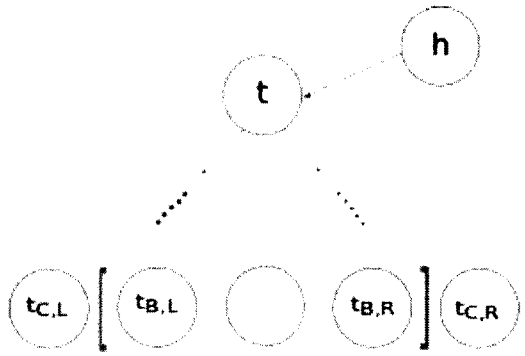


Figure 4-1: A schematic figure illustrating the notion of substitutability in a dependency tree. A word with part-of-speech tag t modifies its head h and projects the constituent span of words between brackets. $t_{B,L}$ and $t_{B,R}$ are the tags of the words at the left and right boundaries of the span; $t_{C,L}$ and $t_{C,R}$ are the tags of the left and right words outside the span.

0.7 and standard deviation to 0.01. The ratio for each sentence is considered a separate draw from the Gaussian distribution.

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

Table 4.2: The manually-specified universal dependency rules used in our experiments. These rules specify head-dependent relationships between coarse syntactic categories.

4.5 Learning and Decoding

Learning and prediction are solved together in our Bayesian model. First, we marginalize out the latent parameters for any setting of the dependency trees $\{y_i\}_{i=1}^N$ in the corpus. This is feasible since the multinomial parameters in our model are all assigned conjugate

symmetric Dirichlet prior distributions. As in the base model, such marginalization results in a joint distribution over $\{y_i\}_{i=1}^N$ and the corresponding sentences (suppressed). We then sample each new tree y_i, one after the other, in the collapsed Gibbs' sampling framework. In other words, a new tree y_i for x_i is sampled from the conditional distribution

$$P(y_i|y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_N, x_1, \ldots, x_N)$$

Note that we do not need the global normalization constant for Gibbs' sampling. However, sampling a tree from this conditional distribution is not typically tractable in our framework. Indeed, some of the overlapping parameters are used more than once in each tree, i.e., they represent counting features. Similarly, our distributional bias towards rule usage does not factor according to the dependency arcs.

We adopt a Metropolis Hastings sampler for each new tree. Our proposal distribution corresponds to a first order parser. It relies on counts of current values of all the other trees (see [46] for details). In order to sample a tree, we first construct an inside table using the proposal distribution (see Appendix A.2). A tree is then sampled in top-down fashion, using marginalized probabilities of all possible trees at lower levels.

In the unsupervised learning setting, we follow the above sampling strategy to update each dependency tree at a time given fixed trees for all the other sentences. In the semi-supervised case, the trees for the annotated part of the data are held fixed to their given gold parses.

Hyper-parameter Settings All the Dirichlet hyper-parameters are set to 0.1 in the unsupervised case and 10 in the semi-supervised case. In the semi-supervised case, the counts from the annotated part of the data are boosted to make that part 10 times bigger than the un-annotated part.

Decoding In the absence of parameters pertaining to non-local decisions, we find and use the tree that maximizes the conditional distribution for each test sentence. When non-local sentence level parameters are present, we use the final MH sample obtained for the test sentence.

4.6 Experimental Setup

Datasets and Evaluation We test the effectiveness of our approach on datasets in 19 languages[2], distributed for the 2006 and 2007 CoNLL shared tasks [14, 63]. Each dataset provides manually annotated dependency trees and part-of-speech tags. To effectively learn from small amounts of data, we also map the gold part-of-speech tags in each corpus to a coarse tagset using the mapping scheme given by Petrov et al. [65].

As the evaluation metric, we use unlabeled directed dependency accuracy. Following standard evaluation practices, for both the baselines and our model we evaluate on sentences of length 50 or less ignoring punctuation.

Training Setup We test our approach in both unsupervised and semi-supervised setups. In the semi-supervised setup, we only use dependency (θ) and constituency (ϕ) parameters [3].

In the semi-supervised case, we experiment with two settings — 10 and 50 annotated sentences. For each setup, the rest of the CoNLL data is used as unparsed text with only part-of-speech tag annotations.

[2] Arabic (ar), Basque (ba), Bulgarian (bu), Catalan (ca), Chinese (ch), Czech (cz), Danish (da), Dutch (du), English (en), German (ge), Greek (gr), Hungarian (hu), Italian (it), Japanese (ja), Portuguese (po), Slovene (sl), Spanish (sp), Swedish (sw) and Turkish (tu)

[3] Our experiments show that in the presence of annotated data, even in small amounts, the rule-based distribution becomes unnecessary.

Baselines We compare the results of our unsupervised experiments against two baselines. The first baseline is the state-of-the-art unsupervised parser of Spitkovsky et al. [76], which is a combination of several existing state-of-the-art parsers. The second baseline is our parsing model from Chapter 2, which incorporates universal rules via posterior regularization. The results reported in Chapter 2 are for sentences up to length 10 and are given only for a subset of the CoNLL languages. We ran the No-Split version of this parser on longer sentences to make the results comparable.

In the semi-supervised case, we compare against a supervised version of our parser which does not have access to additional unlabeled data.

4.7 Results

Unsupervised Results The results of our unsupervised experiments are summarized in Table 4.3. The final two columns show the results of our parser when the basic dependency model is augmented with only rule-based parameters (D1+R) and with both constituency and rule-based parameters (D+C+R). The full version of our model outperforms the rule-based-only one by 3.2% on average. We find some significant differences on certain languages: adding constituency parameters increases performance by 19.1% and 26.8% for Danish and Portuguese respectively, but degrades performance by 8.9% for Hungarian. We speculate that these languages have a relatively complicated relationship between dependency and constituency, and leave a more detailed analysis for future work.

When averaged over all languages, the full version of our unsupervised model gives performance which is 1.9% better than the state-of-the-art parser combination by Spitkovsky et al. [76]; we do better on 7 out of 19 languages. Interestingly, we find considerable differences in performance on individual languages between the two systems. For example,

	COMB	PR	D1+R	D+C+R
ar	26.8	39.4	45.8	51.1
ba	24.4	27.2	34.0	22.7
bu	63.4	55.1	56.5	61.8
ca	68.0	62.9	63.5	63.4
ch	58.4	23.6	51.4	47.8
cz	34.3	39.1	43.2	42.5
da	21.4	41.9	41.2	60.3
du	48.0	54.3	37.1	46.4
en	58.2	42.5	41.9	41.2
ge	56.2	38.9	46.3	49.7
gr	45.4	41.1	58.0	58.0
hu	58.3	50.1	54.8	45.9
it	34.9	57.4	58.0	59.8
ja	63.0	39.7	43.3	51.3
po	74.5	42.3	43.9	70.7
sl	50.9	39.8	46.4	38.9
sp	61.4	54.7	55.7	56.5
sw	49.7	39.9	45.1	52.5
tu	29.2	41.0	36.7	42.3
	48.8	43.7	47.5	50.7

Table 4.3: Unsupervised results, D1 = basic Dependency model, D = all Dependency parameters, C = Constituency parameters and R = Rule-based parameters. COMB and PR refer to the parser COMBination of [76] and the Posterior Regularization-based parser (Chapter 2), respectively.

our system outperforms that of Spitkovsky et al. [76] by 24.3%, 38.9%, 24.9% on Arabic, Danish and Italian, respectively. On the other hand, their system tops ours by 17% on English. Such large differences might indicate that the two systems capture different aspects of the data, which can be benefited from further combination.

We also compare against our parser from Chapter 2, column PR, which uses posterior regularization to incorporate the same set of rules as we do. Current model outperforms by 7% on average and it also gains better results on almost all languages. Even without considering constituency parameters (D1+R), this model achieves a gain of 3.8%. This

indicates that incorporating universal rules as a model parameter is more effective than imposing as posterior constraint. This result can be explained by analyzing the gold tree annotations of our data. The percentage of gold dependencies consistent with rules is a little above 50% when averaged across all languages, with a maximum value of 59% for English. Though we use the same threshold of 0.7 for both our model and PR baseline, our model is more flexible in enforcing the rules. The PR baseline, on the other hand, would try to meet the constraint if possible regardless of the values of underlying model parameters.

Comparison against DMV+CCM (Klein & Manning [47]) We also compare our Multiple views model with the combined DMV and CCM model of Klein & Manning [47]. The dependency and constituency components of our model as well as the datasets we use are different from those of Klein & Manning [47]. To make the comparison fare, we implement a version of our model where the dependency and constituency parameters are the same as in DMV and CCM respectively. Furthermore, we incorporate the smoothing counts used for CCM in the form of parameters of the Dirichlet prior over constituency parameters. This version of our model, when tested on WSJ10 corpus, gives an accuracy of 53%, averaged over 10 runs. The dependency accuracy for the product-of-experts combination of these models as reported in [47] is 47.5%. Our method for combining the two views outperforms this baseline by 5.5%.

Semi-supervised Results Tables 4.4 shows the results for the semi-supervised setups with 10 and 50 annotated sentences. In all cases, the model that combines dependency and constituency views (D+C+Text) outperforms its supervised-only counterpart (D+C) as well as all other model versions, both on average and consistency across (almost) all languages. We see decreasing gains as we increase the amount of annotated sentences available: 4% for 10 sentences and 1% for 50 sentences. Looking at individual languages, we see especially

large gains for Japanese (8.9%) and Chinese (9.1%) with 10 annotated sentences. However, once more annotated data is available, the performance gains are not so drastic.

	with 10 annotated sentences				with 50 annotated sentences			
	D	D+Text	D+C	D+C+Text	D	D+Text	D+C	D+C+Text
ar	62.7	60.8	64.5	67.3	66.2	62.2	70.6	70.3
ba	47.2	46.7	49.1	52.7	55.0	51.9	55.2	56.0
bu	62.6	58.8	62.5	69.1	72.3	72.2	75.9	76.9
ca	70.4	67.5	72.2	77.7	73.5	73.2	78.9	79.8
ch	56.2	66.6	62.1	71.2	74.7	73.5	74.5	78.8
cz	54.0	55.9	60.0	62.0	64.6	60.4	68.6	68.2
da	59.0	59.0	62.2	67.5	67.1	66.6	75.4	76.7
du	40.4	39.9	44.6	52.4	56.1	57.9	61.6	65.0
en	59.1	58.8	66.4	70.1	64.5	61.1	71.0	73.3
ge	62.4	50.4	64.7	67.2	65.2	66.2	71.8	73.1
gr	65.5	65.4	68.0	70.6	69.1	66.1	73.9	74.1
hu	64.3	57.0	65.5	61.3	65.9	63.1	67.8	67.7
it	62.5	61.9	63.0	65.9	65.6	65.3	71.5	70.6
ja	65.5	68.6	65.7	74.6	76.1	78.9	80.4	81.0
po	76.1	75.4	78.2	79.7	78.0	78.5	81.9	82.9
sl	55.5	48.3	60.1	66.1	60.3	54.8	68.1	68.1
sp	61.8	63.2	66.2	69.6	63.9	64.2	73.0	73.4
sw	65.0	56.9	64.4	70.6	69.6	69.5	75.0	76.4
tu	58.9	61.2	60.5	61.8	62.5	64.9	66.9	68.6
	60.5	59.1	63.2	67.2	66.9	65.8	71.7	72.7

Table 4.4: Supervised and semi-supervised results with 10 and 50 supervised sentences, D = Dependency model, C = Constituency parameters; +Text refers to a semi-supervised setup.

Interestingly, we find that semi-supervised learning is only effective in the presence of multiple views. In fact, semi-supervised learning without overlapping views (D+Text) degrades performance by about 1% compared to the supervised version (D). This finding is consistent with previous work [55, 54] where using additional raw text is beneficial for parsing only when the parser is combined with a non-local re-ranker.

Unsupervised Ablation Experiments We also perform ablation experiments to analyze relative contribution of each model component. Table 4.7 shows the results of ablation experiments for different versions of dependency component with or without other views.

	D0	+C	+R	+C+R	D1	+C	+R	+C+R	D	+C	+R	+C+R
ar	33.2	41.5	44.6	45.2	34.6	42.9	45.8	46.5	4.9	39.3	46.6	51.1
ba	23.9	23.2	27.9	28.1	28.8	23.4	34.0	38.4	0.6	21.0	22.6	22.7
bu	12.9	52.1	59.4	59.3	18.3	53.9	56.5	56.0	30.7	58.3	48.5	61.8
ca	26.4	49.5	56.4	25.2	29.3	52.8	63.5	65.5	36.0	36.2	32.6	63.4
ch	72.6	29.8	50.0	43.7	21.6	20.2	51.4	46.8	12.4	21.7	58.2	47.8
cz	31.4	46.3	44.1	39.8	21.3	37.0	43.2	38.3	14.2	25.0	35.9	42.5
da	13.7	38.2	34.9	47.1	17.4	22.3	41.2	53.5	2.4	26.1	44.0	60.3
du	22.5	34.5	30.2	50.3	26.7	34.0	37.1	46.3	16.5	35.6	29.0	46.4
en	15.3	26.9	34.7	36.1	29.0	28.2	41.9	37.1	18.6	29.0	40.5	41.2
ge	28.0	45.5	46.4	43.1	31.1	43.4	46.3	41.3	16.4	33.1	46.7	49.7
gr	31.4	31.4	57.6	58.4	36.9	29.3	58.0	55.7	26.9	33.5	53.0	58.0
hu	48.8	11.2	58.2	32.3	39.5	16.9	54.8	32.8	26.0	24.6	49.3	45.9
it	22.9	50.1	56.5	58.8	24.9	58.8	58.0	55.9	17.8	41.1	52.4	59.8
ja	19.3	22.4	43.3	33.0	57.1	20.9	43.3	33.8	43.2	43.4	53.5	51.3
po	23.0	57.8	44.3	59.5	31.5	69.2	43.9	66.0	10.8	33.5	36.4	70.7
sl	15.9	21.9	34.5	31.5	37.6	36.6	46.4	33.8	20.7	27.7	35.5	38.9
sp	27.0	30.6	55.0	48.9	27.6	42.7	55.7	54.2	31.4	32.9	46.5	56.5
sw	28.0	39.6	46.1	47.5	23.9	39.3	45.1	46.4	12.4	39.8	50.6	52.5
tu	32.5	20.9	40.9	26.8	32.6	16.6	36.7	32.4	25.0	16.2	41.2	42.3
	27.8	35.4	45.5	42.9	30.0	36.2	47.5	46.4	19.3	32.5	43.3	50.7

Table 4.5: Ablation results for unsupervised experiments: D0 = coarse tag Dependency parameters, D1 = fine tag Dependency parameters, D2 = lexicalized Dependency parameters, D = D0+D1+D2 and C = Constituency parameters

In all case adding constituency view or adding universal rules helps. However, adding both of them does not always enhance the performance further except in the case of full dependency component (D+C+R). Moreover, having lexical dependency parameters (D), in the absence of universal rules, always hurts the performance. This is understandable, in the absence of any guidance, bilexical parameters can easily find bad optima.

All versions of our model, when rules are not used, perform worse than that of Spitkovsky

et al. [76]. However, their model employs a more complicated incremental initialization from simpler to more complex versions of the model.

4.8 Conclusions

In this work we introduced a Bayesian model of dependency parsing that operates on the corpus level. The model is built by attaching multiple views to a simple generative model. These views are incorporated as additional parameters that pertain to the same parsing decisions. By integrating over the parameters, we obtain a joint model over the parse trees across the corpus.

Our unsupervised model outperforms a combination of several state-of-the-art parsers [76]. Our method better exploits declarative constraints than posterior regularization, leading to performance improvements. In a semi-supervised setting, the model delivers steady gains from access to raw data.

Chapter 5

Conclusions and Future Work

In this thesis we have shown that declarative linguistic knowledge and linguistically motivated model design can greatly boost parsing performance for low-resource languages.

We present a universal view of dependency parsing. This universal perspective is beneficial in both monolingual and multilingual-transfer setups. In monolingual case, we design a generic parser based on universal rules that consistently improves performance across languages. In the case of crosslingual parser transfer, we combine universal view with linguistic typology. This provides a clean mechanism for filtering-out irrelevant information when transferring from a diverse set of source languages.

We also show that combining multiple linguistic views of data into one model is beneficial in unsupervised and semi-supervised settings. We present a Bayesian framework that can accommodate multiple views in one model without making inference intractable. Our experiments show that when multiple views are merged via this framework, each constrains the others enhancing overall quality of predictions.

5.1 Discussion and Future Work

The work presented in this thesis can be extended in a number of ways:

We explore the idea of separating selection and ordering decisions in the context of parser transfer. This formulation is also supported by linguistic theories [42]. In fact, the unordered dependency tree, labeled with functional roles such as subject and object, can fully convey the meaning. In other words, ordering information is a way of labeling functional roles and different languages can be viewed as different labeling schemes. The use of this two-tier approach for modeling dependencies is worth exploring for monolingual supervised and semi-supervised settings. Furthermore, domain adaptation within language can be viewed as a mirror task of crosslingual transfer, where ordering component is fixed but the selection component may vary at lexical level.

Our corpus level unsupervised model can be enhanced in a number of ways. First, we currently do not make full use of the Bayesian nature of this framework. Especially in the case of rule-based distribution, instead of fixing the parameters, we can use them as priors to allow for variability across languages. Moreover, this framework can be used to capture other non-local overlapping features, such as sibling order and tree-height, without running into the issues of intractability and sparsity. Finally, existing work in NLP has shown that joint learning of multiple tasks improves performance [79, 32]. This framework can be used to learn syntactic structure jointly with other tasks such as morphological analysis and semantic parsing.

Besides the immediate extensions of our work discussed above, a number of question arise from the direction of research followed in this thesis; for instance, how far we can go using available linguistic resources? have we already reached the limit? what would be the logical next step towards enabling syntactic parsing for resource-lean languages?

In Chapter 4 (Section 4.7, Table 4.4), the results show that a parser trained on 10 annotated sentences performs as well as linguistically informed transfer system of Chapter 3. However, annotating first few sentences, for a new language or a new framework, is considered hardest. This is because of the frequent encounter with new types of decisions that require careful analysis. Yet the question remains that in terms of practical usability, would it be a more productive course of action to annotate resources for every language of interest? Especially since a small amount of language specific annotations still outperforms our linguistically based parsers. The solution then may be to adopt a middle approach i.e. to focus on designing parsing methods that specialize in making the most out of a little amount of annotated resources. This was the intent behind the multiple-views semi-supervised experiments of Chapter 4.

This is not to say that we have exhausted the limits of linguistic resources. To the contrary, the work presented in this thesis is just a beginning. There are far more detailed forms of linguistic knowledge that can be incorporated into parsing models such as the Grammar Matrix of Bender et al. [4] and the ParGram Parallel Treebank of Sulger et al. [78]. One hindrance in further progress in this direction is the lack of consistency in annotations against which we evaluate our methods. For instance in our universal rules experiments (Chapter 2) Slovene and Danish data was left out. This was done because of significantly different annotation schemes used in those corpora. The linguistically based models are motivated by unified view of languages as captured by different linguistic theories. Any enhancement in these models will only adhere further to one particular theory/school of thought, making it even harder to evaluate these methods using a set of corpora annotated independently without any regard to consistency[1]. In general, evaluation methods and their limitations have been long standing issues in the area of dependency parsing [68]. An

[1]A recent work by McDonald et al. [56] presents a collection of treebanks for 6 languages that are made consistent in terms of dependency annotations. Currently, 5 out of 6 languages in this collection are Indo-European. However, the authors indicate that this is an on-going effort and the coverage will be improved.

alternative approach can be to resort to application based evaluation i.e. to evaluate the effectiveness of parsing methods based on the improvements they incur to an application system such as Machine Translation. While this alleviate the problem of inconsistency in annotations, there are other issues with this approach. For instance there is no way to distinguish between parser's inability to improve and application's inability to benefit from improvements. Designing evaluation techniques which are not tied to any particular linguistic theory or application software is challenging and constitutes an important research direction.

Finally, linguistically motivated models can also be used to enhance linguistic knowledge. This is discussed briefly in Section 3.6.3, Chapter 3, when analyzing the typological feature weights. It is unfortunate that despite huge improvements in computational capability and increase in the amount of textual resources available in machine readable form, there is relatively little work that uses these resources to unveil linguistic properties of languages [5]. Using computational methods to enhance linguistic knowledge is a line of research that is quite exciting yet less explored.

In conclusion, the work presented in this thesis is only a first step towards exploiting linguistic resources. This direction of research can be followed further to benefit both NLP applications and linguistic knowledge-base.

Appendix A

Inside Algorithms

A.1 Selective Sharing Model

Selection Parameters

$$
\begin{aligned}
P_{size}(n|h) &= \theta_{size}(n|h) \\
P_{set}(S|h,n) &= \frac{1}{Z_{set}(h,n)} e^{\sum_{S_i \in S} \theta_{sel}(S_i|h)} \\
&= \frac{1}{Z_{set}(h,n)} \prod_{S_i \in S} e^{\theta_{sel}(S_i|h)} \\
Z_{set}(h,n) &= \sum_{S:|S|=n} e^{\sum_{S_i \in S} \theta_{sel}(S_i|h)}
\end{aligned}
$$

Where h is a head tag, n is the number of dependents, and S is an unordered set of tags. S_i is the i^{th} POS tag in the unordered set S, and $\theta_{sel}(S_i|h)$ are parameters of the log-linear distribution over sets for h.

Ordering Parameters

$$P_{ord}(d|a,h,l) = \frac{1}{Z_{ord}(a,h,l)} e^{\mathbf{w}_{ord}\cdot\mathbf{g}(d,a,h,\mathbf{v}_l)}$$
$$Z_{ord}(a,h,l) = \sum_{d\in\{R,L\}} e^{\mathbf{w}_{ord}\cdot\mathbf{g}(d,a,h,\mathbf{v}_l)}$$

Where h is a head tag and a is a dependent tag, d is the orientation of the dependency (right or left), and w_{ord} are the weights of ordering log-linear distribution. l is the language index and $\mathbf{v}_l$ is the vector of typological feature values for l.

Recursions for Inside Algorithm: In the following, t_i represents the tag at index i, $\mathcal{R}^n_{i,j}$ represents the sum of the product of selection (un-normalized) and ordering scores of sub-trees rooted at i, spanning from index i to index j, with n dependents on the right, $\mathcal{L}^n_{i,j}$ is the mirror image of $\mathcal{R}^n_{i,j}$ with dependents on the left, and $\mathcal{F}^h_{i,j}$ is the sum of normalized scores of all trees spanning i to j with root at index h. Following formulae give the recursions for $\mathcal{L}$, $\mathcal{R}$ and $\mathcal{F}$, Figure A.1 shows schematic diagram corresponding to each recursion case.

$$\mathcal{R}^n_{i,j} = \sum_{k=i}^{j-1}\sum_{a=k+1}^{j}\left(\mathcal{R}^{n-1}_{i,k}\mathcal{F}^a_{k+1,j}\cdot e^{\theta_{sel}(l_a|t_i)}\cdot P_{ord}(Right|t_a,t_i,l)\right)$$
$$\mathcal{L}^n_{i,j} = \sum_{k=i}^{j-1}\sum_{a=i}^{k}\left(\mathcal{L}^{n-1}_{k+1,j}\mathcal{F}^a_{i,k}\cdot e^{\theta_{sel}(t_a|t_j)}\cdot P_{ord}(Left|t_a,t_j,l)\right)$$
$$\mathcal{F}^h_{i,j} = \sum_{n_l}\sum_{n_r}\left(\mathcal{L}^{n_l}_{i,h}\mathcal{R}^{n_r}_{h,j}\cdot P_{size}(n_l+n_r|t_h)\cdot\frac{1}{Z_{set}(t_h,n_l+n_r)}\cdot\frac{1}{n_l!n_r!}\right)$$

Note that for selection probabilities, we keep multiplying the numerator and normalize only

when the size of the set is known. This works because selection weights depend only on head and dependent tags and they are shared across all set distributions with same head tag.

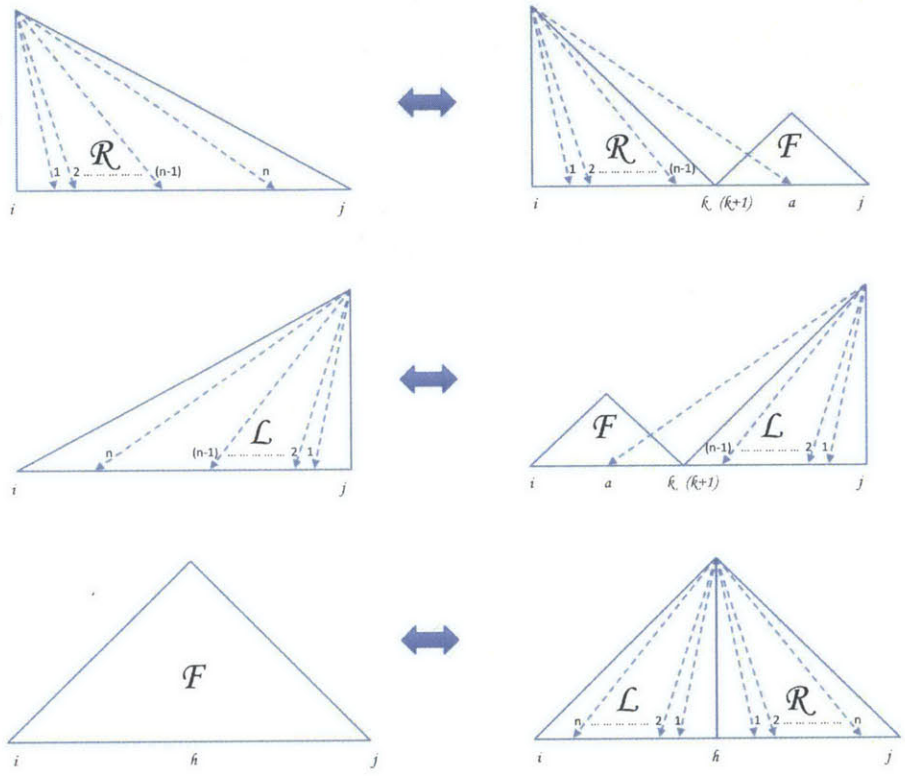


Figure A-1: Schematic diagram of recursions for $\mathcal{L}$, $\mathcal{R}$ and $\mathcal{F}$ cases for inside algorithm

A.2 Multiple Views Model

In this section we explain the inside algorithm for sampling trees for our multiple views model (Chapter 4). The algorithm is described in terms of basic dependency parameters and constituency parameters; other dependency parameters can be added trivially to the algorithm. Rule-based distribution is not a part of the proposal distribution.

In the following, s is the index of the sentence for which we are sampling a new tree and T_{-s} are the current parse trees of all the sentences other than the s_{th} sentence.

Conditional Probabilities for Basic Dependency Decisions

$$P_{stop}(d|h, dir, args, dist, T_{-s}, \theta^0_{stop}) = \frac{Count(d, h, dir, args, dist)_{T_{-s}} + \theta^0_{stop}}{Count(*, h, dir, args, dist)_{T_{-s}} + 2\theta^0_{stop}}$$

$$P_{choose}(a|h, dir, T_{-s}, \theta^0_{choose}) = \frac{Count(a, h, dir)_{T_{-s}} + \theta^0_{choose}}{Count(*, h, dir)_{T_{-s}} + |A| \cdot \theta^0_{choose}}$$

where h is a head tag, a is a dependent tag, dir is the direction of dependency, $dist$ is the number of leaf nodes generated so far on the dir side of head, d represents the binary STOP/GEN decision, θ^0_{stop} and θ^0_{choose} are the parameters of the symmetric Dirichlet priors and A is the set of all POS tags.

Conditional Probabilities for Constituency Decisions

$$P_{const}(t^l_{in}, t^r_{in}|h, g, dir, T_{-s}, \theta^0_{const}) = \frac{Count(t^l_{in}, t^r_{in}, h, g, dir)_{T_{-s}} + \theta^0_{const}}{Count(*, *, h, g, dir)_{T_{-s}} + |B||B|\theta^0_{const}}$$

$$P_{context}(t^l_{out}, t^r_{out}|h, g, dir, T_{-s}, \theta^0_{context}) = \frac{Count(t^l_{out}, t^r_{out}, h, g, dir)_{T_{-s}} + \theta^0_{context}}{Count(*, *, h, g, dir)_{T_{-s}} + |B||B|\theta^0_{context}}$$

where h is a coarse head tag and g is the parent tag of h. dir is the direction of dependency between h and g. t^l_{in} and t^r_{in} are the left and right inner boundary tags of the constituency span projected by h, t^l_{out} and t^r_{out} are the outer boundary tags for the same span. ϕ^0_{const} and $\phi^0_{context}$, are the parameters of the symmetric Dirichlet priors. B is the set of all coarse POS tags.

Recursions for Inside Algorithm: In the following, t_i represents the tag at index i, $\mathcal{R}^n_{i,j}$ represents the sum of scores of subtrees rooted at i, spanning from index i to index j, with n arguments on the right, $\mathcal{L}^n_{i,j}$ is the mirror image of $\mathcal{R}^n_{i,j}$ with dependents on the left, and $\mathcal{F}^h_{i,j}$ is the sum of scores of all trees spanning i to j with root at index h. The function $isarg(a, b)$ has a value of 1 when a is an argument of b and 0 otherwise. This binary decision is based on the rules listed in Table 4.1. Following formulae give the recursions for $\mathcal{L}$, $\mathcal{R}$ and $\mathcal{F}$.

$$\mathcal{R}^n_{i,j} = \sum_{k=i}^{j-1} \sum_{a=k+1}^{j} \Big(\mathcal{R}^{n-isarg(t_a,t_i)}_{i,k} \mathcal{F}^a_{k+1,j}$$
$$\cdot P_{stop}(\textbf{GEN}|t_i, Right, n, k-i, T_{-s})$$
$$\cdot P_{choose}(t_a|t_i, Right, T_{-s})$$
$$\cdot P_{const}(t_{k+1}, t_j|t_a, t_i, Right, T_{-s})$$
$$\cdot P_{context}(t_k, t_{j+1}|t_a, t_i, Right, T_{-s}) \Big)$$

$$\mathcal{L}^n_{i,j} = \sum_{k=i}^{j-1} \sum_{a=i}^{k} \Big(\mathcal{L}^{n-isarg(t_a,t_j)}_{k+1,j} \mathcal{F}^a_{i,k}$$
$$\cdot P_{stop}(\textbf{GEN}|t_j, Left, n, j-k-1, T_{-s})$$
$$\cdot P_{choose}(t_a|t_j, Left, T_{-s})$$
$$\cdot P_{const}(t_i, t_k|t_a, t_j, Left, T_{-s})$$
$$\cdot P_{context}(t_{i-1}, t_{k+1}|t_a, t_j, Left, T_{-s}) \Big)$$

$$\mathcal{F}^h_{i,j} = \sum_{n_l} \sum_{n_r} \Big(\mathcal{L}^{n_l}_{i,h} \mathcal{R}^{n_r}_{h,j}$$
$$\cdot P_{stop}(\textbf{STOP}|t_h, Right, n_r, j-h, T_{-s})$$
$$\cdot P_{stop}(\textbf{STOP}|t_h, Left, n_l, h-i, T_{-s}) \Big)$$

Appendix B

Variational Updates

B.1 Update for $q(\phi_{ts})$

$$\begin{aligned}
\log q(\phi_{ts}) &= E_{q(z)}[\log p(\beta, \pi, \phi, \theta, z, x|\phi_0, \theta_0, \alpha, \gamma)] \\
&\propto E_{q(z)}[\log p(\phi_{ts}|\phi_0)p(x|z, \phi_{ts})] \\
&= \log p(\phi_t|\phi_0) + E_{q(z)}[\log p(x|z, \phi_{ts})] \\
&= \sum_{w\in\Sigma} \log \phi_{ts}(w)^{(\phi_0-1)} + \sum_{w\in\Sigma} E_{q(z)}[\log \phi_{ts}(w)^{C_{ts}(w)}] \\
&\text{where } \Sigma \text{ is the word lexicon} \\
&= \sum_{w\in\Sigma} \log \phi_{ts}(w)^{(\phi_0-1)} + \sum_{w\in\Sigma} E_{q(z)}[C_{ts}(w) \log \phi_{ts}(w)] \\
&= \sum_{w\in\Sigma} \log \phi_{ts}(w)^{(\phi_0-1)} + \sum_{w\in\Sigma} E_{q(z)}[C_{ts}(w)] \log \phi_{ts}(w) \\
&= \sum_{w\in\Sigma} \log \phi_{ts}(w)^{(\phi_0-1)} + \sum_{w\in\Sigma} \log \phi_{ts}(w)^{E_{q(z)}[C_{ts}(w)]} \\
&\text{exponentiating both sides} \\
q(\phi_{ts}) &\propto \prod_{w\in\Sigma} \phi_{ts}(w)^{(\phi_0-1+E_{q(z)}[C_{ts}(w)])} \\
&\propto Dir(\phi_{ts}; \phi_0 + E_{q(z)}[C_{ts}(.)])
\end{aligned}$$

B.2 Update for $q(\pi_{tt's'c})$

$$\begin{aligned}
\log q(\pi_{tt's'c}) &= E_{q(\beta)q(z)}[\log p(\beta, \pi, \phi, \theta, z, x|\phi_0, \theta_0, \alpha, \gamma)] \\
&\propto E_{q(\beta)q(z)}[\log p(\pi_{tt's'c}|\alpha, \beta_t)p(z|\pi_{tt's'c})] \\
&= E_{q(\beta)}[\log p(\pi_{tt's'c}|\alpha, \beta_t)] + E_{q(z)}[\log p(z|\pi_{tt's'c})]
\end{aligned}$$

recall that $q(\beta_t)$ puts all its mass on one β_t

$$\begin{aligned}
&= \log p(\pi_{tt's'c}|\alpha, \beta) + E_{q(z)}[\log p(z|\pi_{tt's'c})] \\
&= \sum_{s=1}^{T} \log \pi_{tt's'c}(s)^{(\alpha\beta_t(s)-1)} + \sum_{s=1}^{T} E_{q(z)}[\log \pi_{tt's'c}(s)^{C_{tt's'c}(s)}] \\
&= \sum_{s=1}^{T} \log \pi_{tt's'c}(s)^{(\alpha\beta_t(s)-1)} + \sum_{s=1}^{T} E_{q(z)}[C_{tt's'c}(s) \log \pi_{tt's'c}(s)] \\
&= \sum_{s=1}^{T} \log \pi_{tt's'c}(s)^{(\alpha\beta_t(s)-1)} + \sum_{s=1}^{T} E_{q(z)}[C_{tt's'c}(s)] \log \pi_{tt's'c}(s) \\
&= \sum_{s=1}^{T} \log \pi_{tt's'c}(s)^{(\alpha\beta_t(s)-1)} + \sum_{s=1}^{T} \log \pi_{tt's'c}(s)^{E_{q(z)}[C_{tt's'c}(s)]}
\end{aligned}$$

exponentiating both sides

$$\begin{aligned}
q(\pi_{tt's'c}) &\propto \prod_{s=1}^{T} \pi_{tt's'c}(s)^{(\alpha\beta_t(s)-1+E_{q(z)}[C_{tt's'c}(s)])} \\
&\propto Dir(\pi_{tt's'c}; \alpha\beta_t(.) + E_{q(z)}[C_{tt's'c}(.)])
\end{aligned}$$

B.3 Update for $q'(z)$

$$
\begin{aligned}
\log q'(z) &= E_{q(\pi),q(\phi),q(\theta)}[\log p(\beta,\pi,\phi,\theta,z,x|\phi_0,\theta_0,\alpha,\gamma)] \\
&\propto E_{q(\pi),q(\phi),q(\theta)}[\log p(x|z,\phi)p(z|\pi,\theta)] \\
&= E_{q(\pi),q(\phi),q(\theta)}\Big[\log \prod_{n=1}^{N}\prod_{i=1}^{l_n}\big(p(x_{ni}|t_{ni},s_{ni},\phi) \times p(t_{ni}|t_{h(ni)},\theta) \\
&\qquad\qquad \times p(s_{ni}|t_{ni},t_{h(ni)},s_{h(ni)},c,\pi)\big)\Big] \\
&= \sum_{s=1}^{N}\sum_{i=1}^{l_n}\Big(E_{q(\phi)}[\log p(x_{ni}|t_{ni},s_{ni},\phi)] \\
&\qquad + E_{q(\theta)}[\log p(t_{ni}|t_{h(ni)},\theta)] \\
&\qquad + E_{q(\pi)}[\log p(s_{ni}|t_{ni},t_{h(ni)},s_{h(ni)},c,\pi)]\Big) \\
&= \sum_{s=1}^{N}\sum_{i=1}^{l_n}\Big(E_{q(\phi)}[\log \phi_{t_{ni},s_{ni}}(x_{ni})] + E_{q(\theta)}[\log \theta_{t_{h(ni)}(t_{ni})}] \\
&\qquad E_{q(\pi)}[\log \pi_{t_{ni},t_{h(ni)},s_{h(ni)},c}(s_{ni})]\Big)
\end{aligned}
$$

exponentiating both sides

$$
\begin{aligned}
q'(z) \propto \prod_{n=1}^{N}\prod_{i=1}^{l_n}\Big(&\exp E_{q(\phi)}[\log \phi_{t_{ni}s_{ni}}(x_{ni})] \\
&\times \exp E_{q(\theta)}[\log \theta_{t_{h(ni)}s_{h(ni)}c_{ni}}(t_{ni})] \\
&\times \exp E_{q(\pi)}[\log \pi_{t_{ni}t_{h(ni)}s_{h(ni)}c_{ni}}(s_{ni})]\Big)
\end{aligned}
$$

B.4 Update for $q(\beta_t)$

$$\begin{aligned}\log q(\beta_t) &\propto E_{q(\pi)}[\log p(\beta_t|\gamma)p(\pi|\alpha,\beta_t)] \\ &= \log p(\beta_t|\gamma) + E_{q(\pi)}[\log \prod_{t'}\prod_{s'}\prod_{c} p(\pi_{tt's'c}|\alpha,\beta_t)] \\ &= \log \text{GEM}(\gamma) + \sum_{t'}\sum_{s'}\sum_{c} E_{q(\pi)}[\log \text{Dir}(\pi_{tt's'c};\alpha\beta_t)]\end{aligned}$$

we use gradient search to find β that maximizes $\log q(\beta_t)$, for details, see Appendix B.4 in [51].

B.5 Variational Bound

$$\begin{aligned}\mathcal{F} &= \int q(\beta,\pi,\theta,\phi,z)\log\frac{p(\beta,\pi,\theta,\phi,z,x|\gamma,\theta_0,\phi_0)}{q(\beta,\pi,\theta,\phi,z)} \\ &= \int q(\beta)q(\pi)q(\theta)q(\phi)q(z)\log\frac{p(\beta|\gamma)p(\pi|\beta)p(\theta|\theta_0)p(\phi|\phi_0)p(z|\pi,\theta)p(x|z,\phi)}{q(\beta)q(\pi)q(\theta)q(\phi)q(z)} \\ &= \int q(z)q(\pi)q(\theta)q(\phi)\log p(z|\pi,\theta)p(x|z,\phi) + \int q(\theta)log\frac{p(\theta|\theta_0)}{q(\theta)} + \int q(\phi)log\frac{p(\phi|\phi_0)}{q(\phi)} \\ &+ \int q(\beta)q(\pi)\log\frac{p(\beta|\gamma)p(\pi|\beta)}{q(\beta)q(\pi)} - \int q(z)\log q(z)\end{aligned}$$

in last term substituting $q(z)$ from Appendix B.3 after normalization

$$= \int q(z)q(\pi)q(\theta)q(\phi)\log p(z|\pi,\theta)p(x|z,\phi) + \int q(\theta)log\frac{p(\theta|\theta_0)}{q(\theta)} + \int q(\phi)log\frac{p(\phi|\phi_0)}{q(\phi)}$$
$$+ \int q(\beta)q(\pi)\log\frac{p(\beta|\gamma)p(\pi|\beta)}{q(\beta)q(\pi)} - \int q(z)E_{q(\pi),q(\theta),q(\phi)}[\log p(z|\pi,\theta)p(x|z,\phi)] + \int q(z)Z_{norm}$$

first and the second last terms cancel each other

$$= Z_{norm} + \int q(\theta)log\frac{p(\theta|\theta_0)}{q(\theta)} + \int q(\phi)log\frac{p(\phi|\phi_0)}{q(\phi)} + \int q(\beta)q(\pi)\log\frac{p(\beta|\gamma)p(\pi|\beta)}{q(\beta)q(\pi)}$$

$q(\beta)$ has probability 1 for β^*

$$= Z_{norm} + \int q(\theta)log\frac{p(\theta|\theta_0)}{q(\theta)} + \int q(\phi)log\frac{p(\phi|\phi_0)}{q(\phi)} + \int q(\pi)\log\frac{p(\beta^*|\gamma)p(\pi|\beta^*)}{q(\pi)}$$
$$= Z_{norm} + \int q(\theta)log\frac{p(\theta|\theta_0)}{q(\theta)} + \int q(\phi)log\frac{p(\phi|\phi_0)}{q(\phi)} + \int q(\pi)\log\frac{p(\pi|\beta^*)}{q(\pi)} + \log p(\beta^*|\gamma)$$
$$= Z_{norm} - KL(q(\theta)||p(\theta|\theta_0)) - KL(q(\phi)||p(\phi|\phi_0)) - KL(q(\pi)||p(\pi|\beta^*)) + \log p(\beta^*|\gamma)$$

where

$$p(\beta^*|\gamma) = GEM(\gamma)$$
$$Z_{norm} = \sum_z E_{q(\pi),q(\theta),q(\phi)}[\log p(z|\pi,\theta)p(x|z,\phi)]$$

KL divergence between two Dirichlet is given by [2]:

$$KL(Dir(\alpha')||Dir(\alpha)) = \ln\frac{\Gamma(\alpha'_0)}{\Gamma(\alpha_0)} - \sum_{j=1}^{K}[\ln\frac{\Gamma(\alpha'_j)}{\Gamma(\alpha_j)} - (\alpha'_j - \alpha_j)(\psi(\alpha'_j) - \psi(\alpha'_0))]$$

where $\alpha_0 = \sum_{j=1}^{K}\alpha_j$

B.6 Gradiant for Dual

We want to find q that optimizes the following objective:

$$\min_q \mathrm{KL}(q||p) \quad \text{s.t.} \quad E_q[f(z)] \leq b$$

The lagrangian for this constrained optimazation has the following form:

$$\begin{aligned}\mathcal{L}(q,\lambda) &= \mathrm{KL}(q||p) + \lambda(E_q[f(z)] - b) \\ &= \sum_z q(z) \log \frac{q(z)}{p(z)} + \lambda\Big(\sum_z q(z) f(z) - b\Big)\end{aligned}$$

taking gradient with respect to $q(z)$,

$$\begin{aligned}\frac{\delta \mathcal{L}(q,\lambda)}{\delta q(z)} &= \log \frac{q(z)}{p(z)} + q(z)\frac{p(z)}{q(z)}\frac{1}{p(z)} + \lambda f(z) \\ &= \log q(z) - \log p(z) + 1 + \lambda f(z)\end{aligned}$$

setting the gradient equal to 0,

$$\begin{aligned}\log q(z) &= \log p(z) - 1 - \lambda f(z) \\ q(z) &= p(z) exp(-1 - \lambda f(z))\end{aligned}$$

to make $q(z)$ sum to 1,

$$q(z) = \frac{p(z) exp(-\lambda f(z))}{Z}$$

substituting $q(z)$ back into $\mathcal{L}$,

$$\begin{aligned}
\mathcal{L}(q,\lambda) &= \sum_z q(z)\log\frac{q(z)}{p(z)} + \lambda\Big(\sum_z q(z)f(z) - b\Big) \\
&= \sum_z q(z)\log\frac{p(z)exp(-\lambda f(z))}{Zp(z)} + \lambda\Big(\sum_z q(z)f(z) - b\Big) \\
&= \sum_z q(z)(-\lambda f(z)) - \sum_z q(z)\log Z + \lambda(E_q[f(z)] - b) \\
&= E_q[-\lambda f(z)] - \log Z + \lambda(E_q[f(z)] - b) \\
&= -\lambda E_q[f(z)] - \log Z + \lambda E_q[f(z)] - \lambda b \\
&= -\log Z - \lambda b
\end{aligned}$$

nowgradient with respect to dual variable λ,

$$\begin{aligned}
\frac{\delta\mathcal{L}(q,\lambda)}{\delta\lambda} &= -\frac{\delta\log Z}{\delta\lambda} - b \\
&= -\sum_z \frac{p(z)exp(-\lambda f(z))(-f(z))}{Z} - b \\
&= -\sum_z q(z)(-f(z)) - b \\
&= E_q[f(z)] - b
\end{aligned}$$

Bibliography

[1] Mark C. Baker. *The Atoms of Language: The Mind's Hidden Rules of Grammar.* Basic Books, 2001.

[2] Matthew J. Beal. *Variational Algorithms for Approximate Bayesian Inference.* PhD thesis, University of London, 2003.

[3] Emily M. Bender. Linguistically naïve != language independent: Why NLP needs linguistic typology. In *Proceedings of the EACL 2009 Workshop on the Interaction between Linguistics and Computational Linguistics: Virtuous, Vicious or Vacuous?*, pages 26–32, 2009.

[4] Emily M. Bender, Dan Flickinger, and Stephan Oepen. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14, 2002.

[5] Emily M Bender and D Terence Langendoen. Computational linguistics in support of linguistic theory. *Linguistic Issues in Language Technology*, 3(1), 2010.

[6] Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. Painless unsupervised learning with features. pages 582–590. Association for Computational Linguistics, 2010.

[7] Taylor Berg-Kirkpatrick and Dan Klein. Phylogenetic grammar induction. In *Proceedings of ACL*, pages 1288–1297, 2010.

[8] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.

[9] Avrim Blum and Tom M. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of COLT*, pages 92–100, 1998.

[10] Phil Blunsom and Trevor Cohn. Unsupervised induction of tree substitution grammars for dependency parsing. In *Proceedings of EMNLP*, pages 1204–1213. Association for Computational Linguistics, 2010.

[11] Prachya Boonkwan and Mark Steedman. Grammar induction from text using small syntactic prototypes. In *IJCNLP*, pages 438–446, 2011.

[12] Prachya Boonkwan and Mark Steedman. Grammar induction from text using small syntactic prototypes. In *Proceedings of IJCNLP*, pages 438–446, 2011.

[13] Jordan L Boyd-Graber and David M Blei. Syntactic topic models. In *Advances in Neural Information Processing Systems*, pages 185–192, 2008.

[14] Sabine Buchholz and Erwin Marsi. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL*, pages 149–164, 2006.

[15] David Burkett and Dan Klein. Two languages are better than one (for syntactic parsing). In *Proceedings of EMNLP*, pages 877–886, 2008.

[16] Andrew Carnie. *Syntax: A Generative Introduction (Introducing Linguistics)*. Blackwell Publishing, 2002.

[17] Ming-Wei Chang, Lev Ratinov, and Dan Roth. Guiding semi-supervision with constraint-driven learning. In *Proceedings of ACL*, pages 280–287, 2007.

[18] Kamalika Chaudhuri, Sham Kakade, Karen Livescu, and Karthik Sridharan. Multi-view clustering via canonical correlation analysis. In *Proceedings of ICML*, 2009.

[19] Shay B. Cohen, Dipanjan Das, and Noah A. Smith. Unsupervised structure prediction with non-parallel multilingual guidance. In *EMNLP*, pages 50–61, 2011.

[20] Shay B. Cohen and Noah A. Smith. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of NAACL/HLT*, pages 74–82, 2009.

[21] Shay B. Cohen and Noah A. Smith. Variational inference for grammar induction with prior knowledge. In *Proceedings of ACL/IJCNLP 2009 Conference Short Papers*, pages 1–4, 2009.

[22] Michael Collins. *Head-driven statistical models for natural language parsing*. PhD thesis, University of Pennsylvania, 1999.

[23] Bernard Comrie. *Language Universals and Linguistic Typology: Syntax and Morphology*. Oxford: Blackwell, 1989.

[24] A. Corduneanu and T. Jaakkola. Continuation methods for mixing heterogeneous sources. In *Proceedings of the Eighteenth Annual Conference on Uncertainty in Artificial Intelligence*, 2002.

[25] Hal Daumé III and Lyle Campbell. A bayesian model for discovering typological implications. In *Proceedings of ACL*, pages 65–72, 2007.

[26] Gregory Druck, Gideon Mann, and Andrew McCallum. Semi-supervised learning of dependency parsers using generalized expectation criteria. In *Proceedings of ACL/IJCNLP*, pages 360–368, 2009.

[27] Jason Eisner and Giorgio Satta. Efficient parsing for bilexical context-free grammars and head automaton grammars. In *Proceedings of ACL*, pages 457–464. Association for Computational Linguistics, 1999.

[28] Jason Eisner and Noah A. Smith. Favor short dependencies: Parsing with soft and hard constraints on dependency length. In *Trends in Parsing Technology: Dependency Parsing, Domain Adaptation, and Deep Parsing*, pages 121–150. 2010.

[29] Richárd Farkas and Bernd Bohnet. Stacking of dependency and phrase structure parsers. In *Proceedings of COLING 2012*, pages 849–866, 2012.

[30] Thomas S. Ferguson. A bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230, 1973.

[31] Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. The infinite tree. In *Proceedings of ACL*, pages 272–279, 2007.

[32] Jenny Rose Finkel and Christopher D Manning. Hierarchical joint learning: Improving joint parsing and named entity recognition with non-jointly labeled data. In *Proceedings of ACL*, pages 720–728, 2010.

[33] Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. Dependency grammar induction via bitext projection constraints. In *Proceedings of ACL/IJCNLP*, pages 369–377, 2009.

[34] Kuzman Ganchev, João Graça, Jennifer Gillenwater, and Ben Taskar. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11:2001–2049, 2010.

[35] Dan Garrette, Jason Mielens, and Jason Baldridge. Real-world semi-supervised learning of pos-taggers for low-resource languages.

[36] Douwe Gelling, Trevor Cohn, Phil Blunsom, and Joao Graça. The pascal challenge on grammar induction. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 64–80. Association for Computational Linguistics, 2012.

[37] João Graça, Kuzman Ganchev, Ben Taskar, and Fernando Pereira. Posterior vs. parameter sparsity in latent variable models. In *Advances in NIPS*, pages 664–672, 2009.

[38] João Graça, Kuzman Ganchev, and Ben Taskar. Expectation maximization and posterior constraints. In *Advances in NIPS*, pages 569–576, 2007.

[39] Nathan Green and Zdeněk Žabokrtský. Hybrid combination of constituency and dependency trees into an ensemble dependency parser. In *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, pages 19–26, 2012.

[40] Joseph H Greenberg. Some universals of language with special reference to the order of meaningful elements. In Joseph H Greenberg, editor, *Universals of Language*, pages 73–113. MIT Press, 1963.

[41] Aria Haghighi and Dan Klein. Prototype-driven grammar induction. In *Proceedings of ACL*, pages 881–888, 2006.

[42] Z.S. Harris. *Mathematical structures of language*. Wiley, 1968.

[43] Martin Haspelmath, Matthew S. Dryer, David Gil, and Bernard Comrie, editors. *The World Atlas of Language Structures*. Oxford University Press, 2005.

[44] William P. Headden III, Mark Johnson, and David McClosky. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of NAACL/HLT*, pages 101–109, 2009.

[45] R. Hwa, P. Resnik, A. Weinberg, C. Cabezas, and O. Kolak. Bootstrapping parsers via syntactic projection across parallel texts. *Journal of Natural Language Engineering*, 11(3):311–325, 2005.

[46] M. Johnson, T. Griffiths, and S. Goldwater. Bayesian Inference for PCFGs via Markov Chain Monte Carlo. In *Proceedings of NAACL-HLT*, pages 139–146, 2007.

[47] Dan Klein and Christopher Manning. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of ACL*, pages 478–485, 2004.

[48] Sandra Kübler, Ryan McDonald, and Joakim Nivre. Dependency parsing. *Synthesis Lectures on Human Language Technologies*, 1(1):1–127, 2009.

[49] Jonas Kuhn. Experiments in parallel-text based grammar induction. In *Proceedings of the ACL*, pages 470–477, 2004.

[50] Percy Liang, Michael I. Jordan, and Dan Klein. Learning from measurements in exponential families. In *Proceedings of ICML*, pages 641–648, 2009.

[51] Percy Liang, Michael I. Jordan, and Dan Klein. Probabilistic grammars and hierarchical Dirichlet processes. *The Handbook of Applied Bayesian Analysis.*, 2009.

[52] Percy Liang, Slav Petrov, Michael Jordan, and Dan Klein. The infinite PCFG using hierarchical Dirichlet processes. In *Proceedings of EMNLP/CoNLL*, pages 688–697, 2007.

[53] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[54] David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Proceedings of HLT-NAACL*, 2006.

[55] David McClosky, Eugene Charniak, and Mark Johnson. Reranking and self-training for parser adaptation. In *Proceedings of ACL*, pages 337–344. Association for Computational Linguistics, 2006.

[56] Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. Universal dependency annotation for multilingual parsing. *Proceedings of ACL, Sofia, Bulgaria*, 2013.

[57] Ryan T. McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. Non-projective dependency parsing using spanning tree algorithms. In *HLT/EMNLP*, 2005.

[58] Ryan T. McDonald, Slav Petrov, and Keith Hall. Multi-source transfer of delexicalized dependency parsers. In *EMNLP*, pages 62–72, 2011.

[59] Tahira Naseem and Regina Barzilay. Using semantic cues to learn syntax. In *AAAI*, 2011.

[60] Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. Using universal linguistic knowledge to guide grammar induction. In *EMNLP*, pages 1234–1244, 2010.

[61] Frederick J. Newmeyer. *Possible and Probable Languages: A Generative Perspective on Linguistic Typology*. Oxford University Press, 2005.

[62] Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932, 2007.

[63] Joakim Nivre, Johan Hall, Sandra Kübler, Ryan T. McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The conll 2007 shared task on dependency parsing. In *EMNLP-CoNLL*, pages 915–932, 2007.

[64] Joakim Nivre and Ryan McDonald. Integrating graph-based and transition-based dependency parsers. In *Proceedings of ACL-08: HLT*, pages 950–958, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[65] Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In *ArXiv*, April 2011.

[66] Slav Petrov and Dan Klein. Learning and inference for hierarchically split PCFGs. In *Proceeding of AAAI*, pages 1663–1666, 2007.

[67] Mohammad Sadegh Rasooli and Heshaam Faili. Fast unsupervised dependency parsing with arc-standard transitions. In *Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP*, pages 1–9, Avignon, France, April 2012. Association for Computational Linguistics.

[68] Roy Schwartz, Omri Abend, Roi Reichart, and Ari Rappoport. Neutralizing linguistically problematic annotations in unsupervised dependency parsing evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 663–672, 2011.

[69] David A. Smith and Noah A. Smith. Bilingual parsing with factored estimation: Using English to parse Korean. In *Proceeding of EMNLP*, pages 49–56, 2004.

[70] Noah A Smith and Jason Eisner. Guiding unsupervised grammar induction using contrastive estimation. In *Proc. of IJCAI Workshop on Grammatical Inference Applications*, pages 73–82, 2005.

[71] Benjamin Snyder, Tahira Naseem, and Regina Barzilay. Unsupervised multilingual grammar induction. In *Proceedings of ACL/AFNLP*, pages 73–81, 2009.

[72] Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay. Adding more languages improves unsupervised multilingual part-of-speech tagging: A bayesian non-parametric approach. In *Proceedings of NAACL-HLT*, pages 83–91. Association for Computational Linguistics, 2009.

[73] Anders Søgaard. Data point selection for cross-language adaptation of dependency parsers. In *ACL (Short Papers)*, pages 682–686, 2011.

[74] Anders Søgaard. Two baselines for unsupervised dependency parsing. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 81–83. Association for Computational Linguistics, 2012.

[75] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. Three dependency-and-boundary models for grammar induction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 688–698, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[76] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. Breaking out of local optima with count transforms and model recombination: A study in grammar induction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1983–1995, Seattle, Washington, USA, October 2013. Association for Computational Linguistics.

[77] Valentin I Spitkovsky, Daniel Jurafsky, and Hiyan Alshawi. Profiting from mark-up: Hyper-text annotations for guided parsing. In *Proceedings of ACL*, pages 1278–1287. Association for Computational Linguistics, 2010.

[78] Sebastian Sulger, Miriam Butt, Tracy Holloway King, Paul Meurer, Tibor Laczkó, György Rákosi, Cheikh Bamba Dione, Helge Dyvik, Victoria Rosén, Koenraad De Smedt, et al. Pargrambank: The pargram parallel treebank.

[79] Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. The conll-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177, 2008.

[80] Oscar Täckström, Ryan McDonald, and Joakim Nivre. Target language adaptation of discriminative transfer parsers. 2013.

[81] Lydia White. *Second Language Acquisition and Universal Grammar.* Cambridge University Press, 2003.

[82] Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403, 1997.

[83] Chenhai Xi and Rebecca Hwa. A backoff model for bootstrapping resources for non-english languages. In *Proceedings of EMNLP*, pages 851 – 858, 2005.

[84] Daniel Zeman and Philip Resnik. Cross-language parser adaptation between related languages. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 35–42, January 2008.

[85] Yuan Zhang, Roi Reichart, Regina Barzilay, and Amir Globerson. Learning to map into a universal pos tagset. In *Proceedings of the 2012 Joint Conference on Empiri-*

cal Methods in Natural Language Processing and Computational Natural Language Learning, pages 1368–1378, 2012.