Improved Parsing and POS Tagging Using Inter-Sentence Consistency Constraints

Alexander M. Rush[1*] **Roi Reichart**[1*] **Michael Collins**[2] **Amir Globerson**[3]

[1]MIT CSAIL, Cambridge, MA, 02139, USA
{srush|roiri}@csail.mit.edu

[2]Department of Computer Science, Columbia University, New-York, NY 10027, USA
mcollins@cs.columbia.edu

[3]School of Computer Science and Engineering, The Hebrew University, Jerusalem, 91904, Israel
gamir@cs.huji.ac.il

Abstract

State-of-the-art statistical parsers and POS taggers perform very well when trained with large amounts of in-domain data. When training data is out-of-domain or limited, accuracy degrades. In this paper, we aim to compensate for the lack of available training data by exploiting similarities between test set sentences. We show how to augment sentence-level models for parsing and POS tagging with inter-sentence consistency constraints. To deal with the resulting global objective, we present an efficient and exact dual decomposition decoding algorithm. In experiments, we add consistency constraints to the MST parser and the Stanford part-of-speech tagger and demonstrate significant error reduction in the domain adaptation and the lightly supervised settings across five languages.

1 Introduction

State-of-the-art statistical parsers and POS taggers perform very well when trained with large amounts of data from their test domain. When training data is out-of-domain or limited, the performance of the resulting model often degrades. In this paper, we aim to compensate for the lack of available training data by exploiting similarities between test set sentences. Most parsing and tagging models are defined at the sentence-level, which makes such inter-sentence information sharing difficult. We show how to augment sentence-level models with inter-sentence constraints to encourage consistent descisions in similar contexts, and we give an efficient algorithm with formal guarantees for decoding such models.

In POS tagging, most taggers perform very well on word types that they have observed in training data, but they perform poorly on unknown words. With a global objective, we can include constraints that encourage a consistent tag across all occurrences of an unknown word type to improve accuracy. In dependency parsing, the parser can benefit from surface-level features of the sentence, but with sparse or out-of-domain training data these features are very noisy. Using a global objective, we can add constraints that encourage similar surface-level contexts to exhibit similar syntactic behaviour.

The first contribution of this work is the use of Markov random fields (MRFs) to model global constraints between sentences in dependency parsing and POS tagging. We represent each word as a node, the tagging or parse decision as its label, and add constraints through edges. MRFs allow us to include global constraints tailored to these problems, and to reason about inference in the corresponding global models.

The second contribution is an efficient dual decomposition algorithm for decoding a global objective with inter-sentence constraints. These constraints generally make direct inference challenging since they tie together the entire test corpus. To alleviate this issue, our algorithm splits the global inference problem into subproblems - decoding of individual sentences, and decoding of the global MRF. These subproblems can be solved efficiently through known methods. We show empirically that by iteratively solving these subproblems, we can find the

[*] Both authors contributed equally to this work.

exact solution to the global model.

We experiment with domain adaptation and lightly supervised training. We demonstrate that global models with consistency constraints can improve upon sentence-level models for dependency parsing and part-of-speech tagging. For domain adaptation, we show an error reduction of up to 7.7% when adapting the second-order projective MST parser (McDonald et al., 2005) from newswire to the QuestionBank domain. For lightly supervised learning, we show an error reduction of up to 12.8% over the same parser for five languages and an error reduction of up to 10.3% over the Stanford trigram tagger (Toutanova et al., 2003) for English POS tagging. The algorithm requires, on average, only 1.7 times the costs of sentence-level inference and finds the exact solution on the vast majority of sentences.

2 Related Work

Methods that combine inter-sentence information with sentence-level algorithms have been applied to a number of NLP tasks. The most similar models to our work are skip-chain CRFs (Sutton and Mccallum, 2004), relational markov networks (Taskar et al., 2002), and collective inference with symmetric clique potentials (Gupta et al., 2010). These models use a linear-chain CRF or MRF objective modified by potentials defined over pairs of nodes or clique templates. The latter model makes use of Lagrangian relaxation. Skip-chain CRFs and collective inference have been applied to problems in IE, and RMNs to named entity recognition (NER) (Bunescu and Mooney, 2004). Finkel et al. (2005) also integrated non-local information into entity annotation algorithms using Gibbs sampling.

Our model can be applied to a variety of off-the-shelf structured prediction models. In particular, we focus on dependency parsing which is characterized by a more complicated structure compared to the IE tasks addressed by previous work.

Another line of work that integrates corpus-level declarative information into sentence-level models includes the posterior regularization (Ganchev et al., 2010; Gillenwater et al., 2010), generalized expectation (Mann and McCallum, 2007; Mann and McCallum,), and Bayesian measurements (Liang et al., 2009) frameworks. The power of these methods has been demonstrated for a variety of NLP tasks, such as unsupervised and semi-supervised POS tagging and parsing. The constraints used by these works differ from ours in that they encourage the posterior label distribution to have desired properties such as sparsity (e.g. a given word can take a small number of labels with a high probability). In addition, these methods use global information during training as opposed to our approach which applies test-time inference global constraints.

The application of dual decomposition for inference in MRFs has been explored by Wainwright et al. (2005), Komodakis et al. (2007), and Globerson and Jaakkola (2007). In NLP, Rush et al. (2010) and Koo et al. (2010) applied dual decomposition to enforce agreement between different sentence-level algorithms for parsing and POS tagging. Work on dual decomposition for NLP is related to the work of Smith and Eisner (2008) who apply belief propagation to inference in dependency parsing, and to constrained conditional models (CCM) (Roth and Yih, 2005) that impose inference-time constraints through an ILP formulation.

Several works have addressed semi-supervised learning for structured prediction, suggesting objectives based on the max-margin principles (Altun and Mcallester, 2005), manifold regularization (Belkin et al., 2005), a structured version of co-training (Brefeld and Scheffer, 2006) and an entropy-based regularizer for CRFs (Wang et al., 2009). The complete literature on domain adaptation is beyond the scope of this paper, but we refer the reader to Blitzer and Daume (2010) for a recent survey.

Specifically for parsing and POS tagging, self-training (Reichart and Rappoport, 2007), co-training (Steedman et al., 2003) and active learning (Hwa, 2004) have been shown useful in the lightly supervised setup. For parser adaptation, self-training (McClosky et al., 2006; McClosky and Charniak, 2008), using weakly annotated data from the target domain (Lease and Charniak, 2005; Rimell and Clark, 2008), ensemble learning (McClosky et al., 2010), hierarchical bayesian models (Finkel and Manning, 2009) and co-training (Sagae and Tsujii, 2007) achieve substantial performance gains. For a recent survey see Plank (2011). Constraints similar to those we use for POS tagging were used by Subramanya et al. (2010) for POS tagger adaptation.

Their work, however, does not show how to decode a global, corpus-level, objective that enforces these constraints, which is a major contribution of this paper.

Inter-sentence syntactic consistency has been explored in the psycholinguistics and NLP literature. Phenomena such as parallelism and syntactic priming – the tendency to repeat recently used syntactic structures – have been demonstrated in human language corpora (e.g. WSJ and Brown) (Dubey et al., 2009) and were shown useful in generative and discriminative parsers (e.g. (Cheung and Penn, 2010)). We complement these works, which focus on consistency between consecutive sentences, and explore corpus level consistency.

3 Structured Models

We begin by introducing notation for sentence-level dependency parsing as a structured prediction problem. The goal of dependency parsing is to find the best parse y for a tagged sentence $x = (w_1/t_1, \ldots, w_n/t_n)$ with words w and POS tags t. Define the *index set* for dependency parsing as

$$\begin{aligned}\mathcal{I}(x) = \{(m,h) \quad : \quad & m \in \{1 \ldots n\}, \\ & h \in \{0 \ldots n\}, m \neq h\}\end{aligned}$$

where $h = 0$ represents the root word. A dependency parse is a vector $y = \{y(m,h) : (m,h) \in \mathcal{I}(x)\}$ where $y(m,h) = 1$ if m is a modifier of the head word h. We define the set $\mathcal{Y}(x) \subset \{0,1\}^{|\mathcal{I}(x)|}$ to be the set of all valid dependency parses for a sentence x. In this work, we use projective dependency parses, but the method also applies to the set of non-projective parse trees.

Additionally, we have a scoring function $f : \mathcal{Y}(x) \to \mathbb{R}$. The optimal parse y^* for a sentence x is given by, $y^* = \arg\max_{y \in \mathcal{Y}(x)} f(y)$. This sentence-level decoding problem can often be solved efficiently. For example in commonly used projective dependency parsing models (McDonald et al., 2005), we can compute y^* efficiently using variants of the Viterbi algorithm.

For this work, we make the assumption that we have an efficient algorithm to find the argmax of

$$f(y) + \sum_{(m,h) \in \mathcal{I}(x)} u(m,h) y(m,h) = f(y) + u \cdot y$$

where u is a vector in $\mathbb{R}^{|\mathcal{I}(x)|}$. In practice, u will be a vector of Lagrange multipliers associated with the dependencies of y in our dual decomposition algorithm given in Section 6.

We can construct a very similar setting for POS tagging where the goal is to find the best tagging y for a sentence $x = (w_1, \ldots, w_n)$. We skip the formal details here.

We next introduce notation for Markov random fields (MRFs) (Koller and Friedman, 2009). An MRF consists of an undirected graph $G = (V, E)$, a set of possible labels for each node L_i for $i \in \{1, \ldots, |V|\}$, and a scoring function g. The index set for MRFs is

$$\begin{aligned}\mathcal{I}^{\text{MRF}} &= \{(i,l) : i \in \{1 \ldots |V|\}, l \in L_i\} \\ &\cup \; \{((i,j), l_i, l_j) : \; (i,j) \in E, l_i \in L_i, l_j \in L_j\}\end{aligned}$$

A label assignment in the MRF is a binary vector z with $z(i,l) = 1$ if the label l is selected at node i and $z((i,j), l_i, l_j) = 1$ if the labels l_i, l_j are selected for the nodes i, j.

In applications such as parsing and POS tagging, some of the label assignments are not allowed. For example, in dependency parsing the resulting structure must be a tree. Consequently, if every node in the MRF corresponds to a word in a document and its label corresponds to the index of its head word, the resulting dependency structure for each sentence must be acyclic. The set of all valid label assignments (one label per node) is given by $\mathcal{Z} \subset \{0,1\}^{|\mathcal{I}^{\text{MRF}}|}$.

We score label assignments in the MRF with a scoring function $g : \mathcal{Z} \to \mathbb{R}$. The best assignment z^* in an MRF is given by, $z^* = \arg\max_{z \in \mathcal{Z}} g(z)$. We focus on pairwise MRFs where this function g is a linear function of z whose parameters are denoted by θ

$$g(z) = z \cdot \theta = \sum_{(i,l) \in \mathcal{I}^{\text{MRF}}} z(i,l)\theta(i,l) + \sum_{((i,j),l_i,l_j) \in \mathcal{I}^{\text{MRF}}} z((i,j), l_i, l_j)\theta((i,j), l_i, l_j)$$

As in parsing, we make the assumption that we have an efficient algorithm to find the argmax of

$$g(z) + \sum_{(i,l) \in \mathcal{I}^{\text{MRF}}(x)} u(i,l) z(i,l)$$

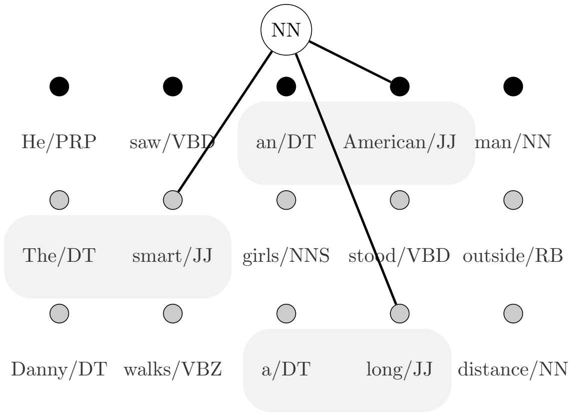

Figure 1: An example constraint from dependency parsing. The black nodes are modifiers observed in the training data. Each gray node corresponds to a possible modifier in the test corpus. The constraint applies to all modifiers in the context `DT JJ`. The white node corresponds to the consensus POS tag of the head word of these modifiers.

4 A Parsing Example

In this section we give a detailed example of global constraints for dependency parsing. The aim is to construct a global objective that encourages similar contexts across the corpus to exhibit similar syntactic behaviour. We implement this objective using an MRF with a node for each word in the test set. The label of each node is the index of the word it modifies. We add edges to this MRF to reward consistency among similar contexts. Furthermore, we add nodes with a fixed label to incorporate contexts seen in the training data.

Specifically, we say that the context of a word is its POS tag and the POS tags of some set of the words around it. We expand on this notion of context in Section 8; for simplicity we assume here that the context includes only the previous word's POS tag. Our constraints are designed to bias words in the same context to modify words with similar POS tags.

Figure 1 shows a global MRF over a small parsing example with one training sentence and two test sentences. The MRF contains a node associated with each word instance, where the label of the node is the index of the word it modifies. In this corpus, the context `DT JJ` appears once in training and twice in testing. We hope to choose head words with similar POS tags for these two test contexts biased by the observed training context.

More concretely, for each context $c \in \{1, \ldots, C\}$, we have a set S_c of associated word indices (s, m) that appear in the context, where s is a sentence index and m is a position in that sentence. For instance, in our example $S_1 = \{(1, 2), (2, 4)\}$ consists of all positions in the test set where we see `JJ` preceded by `DT`. Futhermore, we have a set O_c of indices (s, m, TR) of observed instances of the context in the training data where TR denotes a training index. In our example $O_1 = \{(1, 4, \mathrm{TR})\}$ consists of the one training instance. We associate each word instance with a single context c.

We then define our MRF to include one consensus node for each set S_c as well as a word node for each instance in the set $S_c \cup O_c$. Thus the set of variables corresponds to $V = \{1, \ldots, C\} \cup (\bigcup_{c=1}^{C} S_c \cup O_c)$. Additionally, we include an edge from each node $i \in S_c \cup O_c$ to its consensus node c, $E = \{(i, c) : c \in \{1, \ldots, C\}, i \in S_c \cup O_c\}$. The word nodes from S_c have the label set of possible head indices $L_{(s,m)} = \{0, \ldots, n_s\}$ where n_s is the length of the sentence s. The observed nodes from O_c have a singleton label set $L_{(s,m,\mathrm{TR})}$ with the observed index. The consensus nodes have the label set $L_c = \mathcal{T} \cup \{\mathrm{NULL}\}$ where $\mathcal{T}$ is the set of POS tags and the NULL symbol represents the constraint being turned off.

We can now define the scoring function g for this MRF. The scoring function aims to reward consistency among the head POS tag at each word and the consensus node

$$\theta((i, c), l_i, l_c) = \begin{cases} \delta_1 & \text{if } pos(l_i) = l_c \\ \delta_2 & \text{if } pos(l_i) \text{ is close to } l_c \\ \delta_3 & l_c = \mathrm{NULL} \\ 0 & \text{otherwise} \end{cases}$$

where pos maps a word index to its POS tag. The parameters $\delta_1 \geq \delta_2 \geq \delta_3 \geq 0$ determine the bonus for identical POS tags, similar POS tags, and for turning off the constraint .

We construct a similar model for POS tagging. We choose sets T_c corresponding to the c'th unknown word type in the corpus. The MRF graph is identical to the parsing case with T_c replacing S_c and we no longer have O_c. The label sets for the word nodes are now $L_{(s,m)} = \mathcal{T}$ where the label is the POS tag chosen at that word, and the label set for the consensus node is $L_c = \mathcal{T} \cup \{\text{NULL}\}$. We use the same scoring function as in parsing to enforce consistency between word nodes and the consensus node.

5 Global Objective

Recall the definition of sentence-level parsing, where the optimal parse y^* for a single sentence x under a scoring function f is given by: $y^* = \arg\max_{y \in \mathcal{Y}(x)} f(y)$. We apply this objective to a set of sentences, specified by the tuple $X = (x_1, ..., x_r)$, and the product of possible parses $\mathcal{Y}(X) = \mathcal{Y}(x_1) \times \ldots \times \mathcal{Y}(x_r)$. The sentence-level decoding problem is to find the optimal dependency parses $Y^* = (Y_1^*, ..., Y_r^*) \in \mathcal{Y}(X)$ under a global objective

$$Y^* = \arg\max_{Y \in \mathcal{Y}(X)} F(Y) = \arg\max_{Y \in \mathcal{Y}(X)} \sum_{s=1}^{r} f(Y_s)$$

where $F : \mathcal{Y}(X) \rightarrow \mathbb{R}$ is the global scoring function.

We now consider scoring functions where the global objective includes inter-sentence constraints. Objectives of this form will not factor directly into individual parsing problems; however, we can choose to write them as the sum of two convenient terms: (1) A simple sum of sentence-level objectives; and (2) A global MRF that connects the local structures.

For convenience, we define the following index set.

$$\mathcal{J}(X) = \{(s, m, h) : \quad s \in \{1, \ldots, r\}, \\ (m, h) \in \mathcal{I}(x_s)\}$$

This set enumerates all possible dependencies at each sentence in the corpus. We say the parses Y_s are consistent with a label assignment z if for all $(s, m, h) \in \mathcal{J}(X)$ we have that $z((s, m), h) = Y_s(m, h)$. In other words, the labels in z match the head words chosen in parse Y_s.

With this notation we can write the full global decoding objective as

$$(Y^*, z^*) = \arg\max_{Y \in \mathcal{Y}(X),\, z \in \mathcal{Z}} F(Y) + g(z) \qquad (1)$$

$$\text{s.t. } \forall (s, m, h) \in \mathcal{J}(X),\; z((s, m), h) = Y_s(m, h)$$

```
Set $u^{(1)}(s,m,h) \leftarrow 0$ for all $(s,m,h) \in \mathcal{J}(X)$
for $k = 1$ to $K$ do
    $z^{(k)} \leftarrow \arg\max_{z \in \mathcal{Z}} \; (g(z) + \sum_{(s,m,h) \in \mathcal{J}(X)} u^{(k)}(s,m,h) z((s,m),h))$
    $Y^{(k)} \leftarrow \arg\max_{Y \in \mathcal{Y}(X)} (F(Y) - \sum_{(s,m,h) \in \mathcal{J}(X)} u^{(k)}(s,m,h) Y_s(m,h))$
    if $Y_s^{(k)}(m,h) = z^{(k)}((s,m),h)$
        for all $(s,m,h) \in \mathcal{J}(X)$ then
        return $(Y^{(k)}, z^{(k)})$
    for all $(s,m,h) \in \mathcal{J}(X)$,
        $u^{(k+1)}(s,m,h) \leftarrow u^{(k)}(s,m,h) + \alpha_k(z^{(k)}((s,m),h) - Y_s^{(k)}(m,h))$
return $(Y^{(K)}, z^{(K)})$
```

Figure 2: The global decoding algorithm for dependency parsing models.

The solution to this objective maximizes the local models as well as the global MRF, while maintaining consistency among the models. Specifically, the MRF we use in the experiments has a simple naive Bayes structure with the consensus node connected to all relevant word nodes.

The global objective for POS tagging has a similar form. As before we add a node to the MRF for each word in the corpus. We use the POS tag set as our labels for each of these nodes. The index set contains an element for each possible tag at each word instance in the corpus.

6 A Global Decoding Algorithm

We now consider the decoding question: how to find the structure Y^* that maximizes the global objective. We aim for an efficient solution that makes use of the individual solvers at the sentence-level. For this work, we make the assumption that the graph chosen for the MRF has small tree-width, e.g. our naive Bayes constraints, and can be solved efficiently using dynamic programming.

Before we describe our dual decomposition algorithm, we consider the difficulty of solving the global objective directly. We have an efficient dynamic programming algorithm for solving dependency parsing at the sentence-level, and efficient algorithms for solving the MRF. It follows that we could construct an intersected dynamic programming algorithm that maintains the product of states over both models. This algorithm is exact, but it is very inefficient. Solving the intersected dynamic program requires decoding simultaneously over the entire corpus, with an additional multiplicative factor for solving the MRF. On top of this cost, we need to alter the internal structure of the sentence-level models.

In contrast, we can construct a dual decomposition algorithm which is efficient, produces a certificate when it finds an exact solution, and directly uses the sentence-level parsing models. Considering again the global objective of equation 1, we note that the difficulty in decoding this objective comes entirely from the constraints $z((s,m),h) = Y_s(m,h)$. If these were not there, the problem would factor into two parts, an optimization of F over the test corpus $\mathcal{Y}(X)$ and an optimization of g over possible MRF assignments $\mathcal{Z}$. The first problem factors naturally into sentence-level parsing problems and the second can be solved efficiently given our assumptions on the MRF topology G.

Recent work has shown that a relaxation based on dual decomposition often produces an exact solution for such problems (Koo et al., 2010). To apply dual decomposition, we introduce Lagrange multipliers $u(s,m,h)$ for the agreement constraints between the sentence-level models and the global MRF. The Lagrangian dual is the function $L(u) = \max_z g(z,u) + \max_y F(y,u)$ where

$$g(z,u) = g(z) \; + \sum_{(s,m,h) \in \mathcal{J}(X)} u(s,m,h) z((s,m),h)),$$

$$F(y,u) = F(Y) \; - \sum_{(s,m,h) \in \mathcal{J}(X)} u(s,m,h) Y_s(m,h)$$

In order to find $\min_u L(u)$, we use subgradient descent. This requires computing $g(z,u)$ and $F(y,u)$ for fixed values of u, which by our assumptions from Section 3 are efficient to calculate.

The full algorithm is given in Figure 2. We start with the values of u initialized to 0. At each iteration k, we find the best set of parses $Y^{(k)}$ over the entire corpus and the best MRF assignment $z^{(k)}$. We then update the value of u based on the difference between $Y^{(k)}$ and $z^{(k)}$ and a rate parameter α. On the next iteration, we solve the same decoding problems modified by the new value of u. If at any point the current solutions $Y^{(k)}$ and $z^{(k)}$ satisfy the consistency constraint, we return their current values. Otherwise, we stop at a max iteration K and return the values from the last iteration.

	$\geq$ 0.7	$\geq$0.8	$\geq$ 0.9	1.0
All Contexts	66.8	57.9	46.8	33.3
Head in Context	76.0	67.9	57.2	42.3

Table 1: Exploratory statistics for constraint selection. The table shows the percentage of context types for which the probability of the most frequent head tag is at least p. Head in Context refers to the subset of contexts where the most frequent head is within the context itself. Numbers are based on Section 22 of the Wall Street Journal and are given for contexts that appear at least 10 times.

We now give a theorem for the formal guarantees of this algorithm.

Theorem 1 *If for some $k \in \{1 \ldots K\}$ in the algorithm in Figure 2, $Y_s^{(k)}(m,h) = z^{(k)}(s,m,h)$ for all $(s,m,h) \in \mathcal{J}$, then $(Y^{(k)}, z^{(k)})$ is a solution to the maximization problem in equation 1.*

We omit the proof for brevity. It is a slight variation of the proof given by Rush et al. (2010).

7 Consistency Constraints

In this section we describe the consistency constraints used for the global models of parsing and tagging.

Parsing Constraints. Recall from Section 4 that we choose parsing constraints based on the word context. We encourage words in similar contexts to choose head words with similar POS tags.

We use a simple procedure to select which constraints to add. First define a context template to be a set of offsets $\{r, \ldots, s\}$ with $r \leq 0 \leq s$ that specify the neighboring words to include in a context. In the example of Figure 1, the context template $\{-1, 0, 1, 2\}$ applied to the word `girls/NNS` would produce the context `JJ NNS VBD RB`. For each word in the corpus, we consider all possible templates with $s - r < 4$. We use only contexts that predict the head POS of the context in the training data with probability 1 and prefer long over short contexts. Once we select the context of each word, we add a consensus node for each context type in the corpus. We connect each word node to its corresponding consensus node.

Local context does not fully determine the POS tag of the head word, but for certain contexts it provides a strong signal. Table 1 shows context statistics for English. For 46.8% of the contexts, the most frequent head tag is chosen $\geq 90\%$ of the time. The pattern is even stronger for contexts where the most frequent head tag is within the context itself. In this case, for 57.2% of the contexts the most frequent head tag is chosen $\geq 90\%$ of the time. Consequently, if more than one context can be selected for a word, we favor the contexts where the most frequent head POS is inside the context.

POS Tagging Constraints. For POS tagging, our constraints focus on words not observed in the training data. It is well-known that each word type appears only with a small number of POS tags. In Section 22 of the WSJ corpus, 96.35% of word types appear with a single POS tag.

In most test sets we are unlikely to see an unknown word more than once or twice. To fix this sparsity issue, we import additional unannotated sentences for each unknown word from the New York Times Section of the NANC corpus (Graff, 1995). These sentences give additional information for unknown word types.

Additionally, we note that morphologically related words often have similar POS tags. We can exploit this relationship by connecting related word types to the same consensus node. We experimented with various morphological variants and found that connecting a word type with the type generated by appending the suffix “s” was most beneficial. For each unknown word type, we also import sentences for its morphologically related words.

8 Experiments and Results

We experiment in two common scenarios where parsing performance is reduced from the fully supervised, in-domain case. In domain adaptation, we train our model completely in one source domain and test it on a different target domain. In lightly supervised training, we simulate the case where only a limited amount of annotated data is available for a language.

	Base	ST	Model	ER
WSJ → QTB	89.63	89.99	90.43	7.7
QTB → WSJ	74.89	74.97	75.76	3.5

Table 2: Dependency parsing UAS for domain adaptation. WSJ is the Penn TreeBank. QTB is the QuestionBank. ER is error reduction. Results are significant using the sign test with $p \leq 0.05$.

Data for Domain Adaptation We perform domain adaptation experiments in English using the WSJ PennTreebank (Marcus et al., 1993) and the QuestionBank (QTB) (Judge et al., 2006). In the WSJ → QTB scenario, we train on sections 2-21 of the WSJ and test on the entire QTB (4000 questions). In the QTB → WSJ scenario, we train on the entire QTB and test on section 23 of the WSJ.

Data for Lightly Supervised Training For all English experiments, our data was taken from the WSJ PennTreebank: training sentences from Section 0, development sentences from Section 22, and test sentences from Section 23. For experiments in Bulgarian, German, Japanese, and Spanish, we use the CONLL-X data set (Buchholz and Marsi, 2006) with training data taken from the official training files. We trained the sentence-level models with 50-500 sentences. To verify the robustness of our results, our test sets consist of the official test sets augmented with additional sentences from the official training files such that each test file consists of 25,000 words. Our results on the official test sets are very similar to the results we report and are omitted for brevity.

Parameters The model parameters, δ_1, δ_2, and δ_3 of the scoring function (Section 4) and α of the Lagrange multipliers update rule (Section 6), were tuned on the English development data. In our dual decomposition inference algorithm, we use $K = 200$ maximum iterations and tune the decay rate following the protocol described by Koo et al. (2010).

Sentence-Level Models For dependency parsing we utilize the second-order projective MST parser (McDonald et al., 2005)[1] with the gold-standard POS tags of the corpus. For POS tagging we use the Stanford POS tagger (Toutanova et al., 2003)[2].

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

[2] http://nlp.stanford.edu/software/tagger.shtml

	50			100			200			500		
	Base	ST	Model (ER)	Base	ST	Model (ER)	Base	ST	Model (ER)	Base	ST	Model (ER)
Jap	79.10	80.19	81.78 (12.82)	81.53	81.59	83.09 (8.45)	84.84	85.05	85.50 (4.35)	87.14	87.24	87.44 (2.33)
Eng	69.60	69.73	71.62 (6.64)	73.97	74.01	75.27 (4.99)	77.67	77.68	78.69 (4.57)	81.83	81.90	82.18 (1.93)
Spa	71.67	71.72	73.19 (5.37)	74.53	74.63	75.41 (3.46)	77.11	77.09	77.44 (1.44)	79.97	79.88	80.04 (0.35)
Bul	71.10	70.59	72.13 (3.56)	73.35	72.96	74.61 (4.73)	75.38	75.54	76.17 (3.21)	81.95	81.75	82.18 (1.27)
Ger	68.21	68.28	68.83 (1.95)	72.19	72.29	72.76 (2.05)	74.34	74.45	74.95 (2.4)	77.20	77.09	77.51 (1.4)

Table 3: Dependency parsing UAS by size of training set and language. English data is from the WSJ. Bulgarian, German, Japanese, and Spanish data is from the CONLL-X data sets. Base is the second-order, projective dependency parser of McDonald et al. (2005). ST is a self-training model based on Reichart and Rappoport (2007). Model is the same parser augmented with inter-sentence constraints. ER is error reduction. Using the sign test with $p \leq 0.05$, all 50, 100, and 200 results are significant, as are Eng and Ger 500.

	50		100		200		500	
	Base	Model (ER)	Base	Model (ER)	Base	Model (ER)	Base	Model (ER)
Acc	79.67	81.77 (10.33)	85.42	86.37 (6.52)	88.63	89.37 (6.51)	91.59	91.98 (4.64)
Unk	62.88	67.16 (11.53)	71.10	73.32 (7.68)	75.82	78.07 (9.31)	80.67	82.28 (8.33)

Table 4: POS tagging accuracy. Stanford POS tagger refers to the maximum entropy trigram tagger of Toutanova et al. (2003). Our inter-sentence POS tagger augments this baseline with global constraints. ER is error reduction. All results are significant using the sign test with $p \leq 0.05$.

Evaluation and Baselines To measure parsing performance, we use unlabeled attachment score (UAS) given by the CONLL-X dependency parsing shared task evaluation script (Buchholz and Marsi, 2006). We compare the accuracy of dependency parsing with global constraints to the sentence-level dependency parser of McDonald et al. (2005) and to a self-training baseline (Steedman et al., 2003; Reichart and Rappoport, 2007). The parsing baseline is equivalent to a single round of dual decomposition. For the self-training baseline, we parse the test corpus, append the labeled test sentences to the training corpus, train a new parser, and then re-parse the test set. We run this procedure for a single iteration.

For POS tagging we measure token level POS accuracy for all the words in the corpus and also for unknown words (words not observed in the training data). We compare the accuracy of POS tagging with global constraints to the accuracy of the Stanford POS tagger [3].

Domain Adaptation Accuracy Results are presented in Table 2. The constrained model reduces the error of the baseline on both cases. Note that when the base parser is trained on the WSJ corpus its UAS performance on the QTB is 89.63%. Yet, the constrained model is still able to reduce the baseline error by 7.7%.

Lightly Supervised Accuracy The parsing results are given in Table 3. Our model improves over the baseline parser and self-training across all languages and training set sizes. The best results are for Japanese and English with error reductions of 2.33 – 12.82% and 1.93 – 6.64% respectively. The self-training baseline achieves small gains on some languages, but generally performs similarly to the standard parser.

The POS tagging results are given in Table 4. Our model improves over the baseline tagger for the entire training size range. For 50 training sentences we reduce 10.33% of the overall error, and 11.53% of the error on unknown words. Although the tagger performance substantially improves when the training set grows to 500 sentences, our model still provides an overall error reduction of 4.64% and of 8.33% for unknown words.

9 Discussion

Efficiency Since dual decomposition often requires hundreds of iterations to converge, a naive implementation would be orders of magnitude slower than the underlying sentence-level model. We use two techniques to speed-up the algorithm.

First, we follow Koo et al. (2010) and use lazy decoding as part of dual decomposition. At each iteration k, we cache the result of the MRF $z^{(k)}$ and set of parse tree $Y^{(k)}$. In the next iteration, we only

[3] We do not run self-training for POS tagging as it has been shown unuseful for this application (Clark et al., 2003).

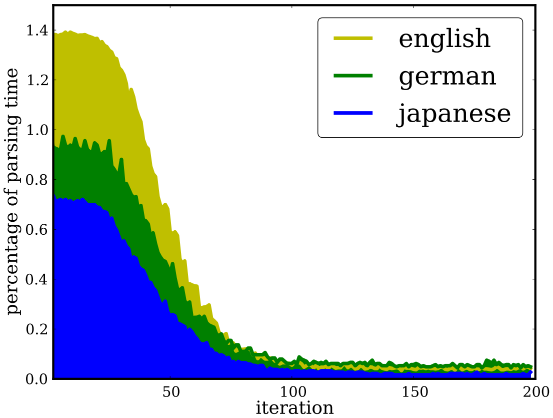

Figure 3: Efficiency of dependency parsing decoding for three languages. The plot shows the speed of each iteration of the subgradient algorithm relative to a round of unconstrained parsing.

Most Effective Contexts	
WSJ → QTB	QTB → WSJ
WRB VBP	VBD NN **NN** ,
DT JJS NN **IN**	IN PRP **VBZ**
VBP PRP **VB**	**JJ** JJ NN ,
DT **NN** NN VB	IN **JJ** JJ NN
RBS JJ NN **IN**	**NN** POS NN NN

Table 5: The five most effective constraint contexts from the domain adaptation experiments. The bold POS tag indicates the modifier word of the context.

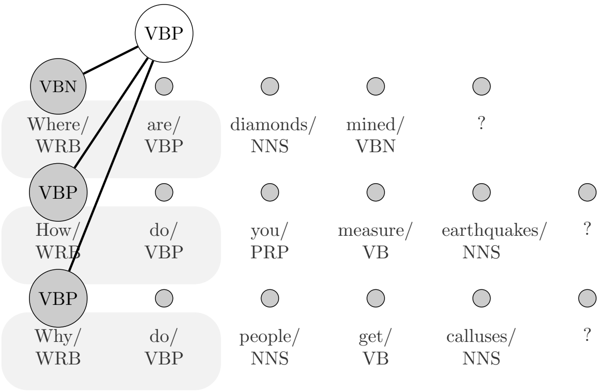

Figure 4: Subset of sentences with the context **WRB** VBP from WSJ → QTB domain adaptation. In the first round, the parser chooses VBN for the first sentence, which is inconsistent with similar contexts. The constraints correct this choice in later rounds.

recompute the solution Y_s^* for a sentence s if the weight $u(s, m, h)$ for some m, h was updated. A similar technique is applied to the MRF.

Second, during the first iteration of the algorithm we apply max-marginal based pruning using the threshold defined by Weiss and Taskar (2010). This produces a pruned hypergraph for each sentence, which allows us to avoid recomputing parse features and to solve a simplified search problem.

To measure efficiency, we compare the time spent in dual decomposition to the speed of unconstrained inference. Across experiments, the mean dual decomposition time is 1.71 times the cost of unconstrained inference. Figure 3 shows how this time is spent after the first iteration. The early iterations are around 1% of the total cost, and because of lazy decoding this quickly drops to almost nothing.

Exactness To measure exactness, we count the number of sentences for which we should remove the constraints in order for the model to reach convergence. For dependency parsing, across languages removing constraints on 0.6% of sentences yields exact convergence. Removing these constraints has very little effect on the final outcome of the model. For POS tagging, the algorithm finds an exact solution after removing constraints from 0.2% of the sentences.

Constraint Analysis We can also look at the number, size, and outcome of the constraints chosen in the experiments. In the lightly supervised experiments, the average number of constraints is 3298 for 25000 tokens, where the median constraint connects 19 different tokens. Of these constraints around 70% are active (non-NULL). The domain adaptation experiments have a similar number of constraints with around 75% of constraints active. In both experiments many of the constraints are found to be consistent after the first iteration, but as Figure 3 implies, other constraints take multiple iterations to converge.

Qualitative Analysis In order to understand why these simple consistency constraints are effective, we take a qualitative look at the the domain adaptation experiments on the QuestionBank. Table 5 ranks the five most effective contextual constraints from both experiments. For the WSJ → QTB experiment, the most effective constraint relates the inital question word with an adjacent verb. Figure 4 shows sentences where this constraint applies in the QuestionBank. For the QTB → WSJ experiment, the effective contexts are mostly long base noun phrases. These occur often in the WSJ but are rare in the simpler QuestionBank sentences.

10 Conclusion

In this work we experiment with inter-sentence consistency constraints for dependency parsing and POS tagging. We have proposed a corpus-level objective that augments sentence-level models with such constraints and described an exact and efficient dual decomposition algorithm for its decoding. In future work, we intend to explore efficient techniques for joint parameter learning for both the global MRF and the local models.

Acknowledgments Columbia University gratefully acknowledges the support of the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of DARPA, AFRL, or the US government. Alexander Rush was supported by a National Science Foundation Graduate Research Fellowship.

References

Y. Altun and D. Mcallester. 2005. Maximum margin semi-supervised learning for structured variables. In *NIPS*.

M. Belkin, P. Niyogi, and V. Sindhwani. 2005. On manifold regularization. In *AISTATS*.

John Blitzer and Hal Daume. 2010. Icml 2010 tutorial on domain adaptation. In *ICML*.

U. Brefeld and T. Scheffer. 2006. Semi-supervised learning for structured output variables. In *ICML*.

S. Buchholz and E. Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *CoNLL*.

R.C. Bunescu and R.J. Mooney. 2004. Collective information extraction with relational markov networks. In *ACL*.

J.C.K Cheung and G. Penn. 2010. Utilizing extra-sentential context for parsing. In *EMNLP*.

Stephen Clark, James Curran, and Miles Osborne. 2003. Bootstrapping pos taggers using unlabelled data. In *CoNLL*.

A. Dubey, F. Keller, and P. Sturt. 2009. A probabilistic corpus-based model of parallelism. *Cognition*, 109(2):193–210.

Jenny Rose Finkel and Christopher Manning. 2009. Hierarchical bayesian domain adaptation. In *NAACL*.

J.R. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*.

K. Ganchev, J. Graça, J. Gillenwater, and B. Taskar. 2010. Posterior Regularization for Structured Latent Variable Models. *Journal of Machine Learning Research*, 11:2001–2049.

J. Gillenwater, K. Ganchev, J. Graça, F. Pereira, and B. Taskar. 2010. Sparsity in dependency grammar induction. In *Proceedings of the ACL Conference Short Papers*.

A. Globerson and T. Jaakkola. 2007. Fixing max-product: Convergent message passing algorithms for map lp-relaxations. In *NIPS*.

D. Graff. 1995. North american news text corpus. *Linguistic Data Consortium*, LDC95T21.

Rahul Gupta, Sunita Sarawagi, and Ajit A. Diwan. 2010. Collective inference for extraction mrfs coupled with symmetric clique potentials. *JMLR*.

R. Hwa. 2004. Sample selection for statistical parsing. *Computational Linguistics*, 30(3):253–276.

John Judge, Aoife Cahill, and Josef van Genabith. 2006. Questionbank: Creating a corpus of parse-annotated questions. In *ACL-COLING*.

D. Koller and N. Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

N. Komodakis, N. Paragios, and G. Tziritas. 2007. MRF optimization via dual decomposition: Message-passing revisited. In *ICCV*.

T. Koo, A.M. Rush, M. Collins, T. Jaakkola, and D. Sontag. 2010. Dual decomposition for parsing with non-projective head automata. In *EMNLP*.

Matthew Lease and Eugene Charniak. 2005. Parsing biomedical literature. In *IJCNLP*.

P. Liang, M. I. Jordan, and D. Klein. 2009. Learning from measurements in exponential families. In *ICML*.

G.S. Mann and A. McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of Machine Learning Research*, 11:955–984.

G.S. Mann and A. McCallum. 2007. Simple, robust, scalable semi-supervised learning via expectation regularization. In *ICML*.

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

David McClosky and Eugene Charniak. 2008. Self-training for biomedical parsing. In *ACL, sort papers*.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Reranking and self-training for parser adaptation. In *ACL*.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adapatation for parsing. In *NAACL*.

R.T. McDonald, F. Pereira, K. Ribarov, and J. Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *HLT/EMNLP*.

Barbara Plank. 2011. *Domain Adaptation for Parsing*. Ph.d. thesis, University of Groningen.

R. Reichart and A. Rappoport. 2007. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *ACL*.

Laura Rimell and Stephen Clark. 2008. Adapting a lexicalized-grammar parser to contrasting domains. In *EMNLP*.

D. Roth and W. Yih. 2005. Integer linear programming inference for conditional random fields. In *ICML*.

A.M. Rush, D. Sontag, M. Collins, and T. Jaakkola. 2010. On dual decomposition and linear programming relaxations for natural language processing. In *EMNLP*.

Kenji Sagae and Junichi Tsujii. 2007. Dependency parsing and domain adaptation with lr models and parser ensembles. In *EMNLP-CoNLL*.

D.A. Smith and J. Eisner. 2008. Dependency parsing by belief propagation. In *EMNLP*.

M. Steedman, M. Osborne, A. Sarkar, S. Clark, R. Hwa, J. Hockenmaier, P. Ruhlen, S. Baker, and J. Crim. 2003. Bootstrapping statistical parsers from small datasets. In *EACL*.

Amarnag Subramanya, Slav Petrov, and Fernando Pereira. 2010. Efficient graph-based semi-supervised learning of structured tagging models. In *EMNLP*.

C. Sutton and A. Mccallum. 2004. Collective segmentation and labeling of distant entities in information extraction. In *In ICML Workshop on Statistical Relational Learning and Its Connections*.

B. Taskar, P. Abbeel, and d. Koller. 2002. Discriminative probabilistic models for relational data. In *UAI*.

K. Toutanova, D. Klein, C.D. Manning, and Y. Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *HLT-NAACL*.

M. Wainwright, T. Jaakkola, and A. Willsky. 2005. MAP estimation via agreement on trees: message-passing and linear programming. In *IEEE Transactions on Information Theory*, volume 51, pages 3697–3717.

Y. Wang, G. Haffari, S. Wang, and G. Mori. 2009. A rate distortion approach for semi-supervised conditional random fields. In *NIPS*.

D. Weiss and B. Taskar. 2010. Structured prediction cascades. In *Proc. of AISTATS*, volume 1284.