Neutralizing Linguistically Problematic Annotations in Unsupervised Dependency Parsing Evaluation

Roy Schwartz[1] **Omri Abend**[1*] **Roi Reichart**[2] **Ari Rappoport**[1]

[1]Institute of Computer Science
Hebrew University of Jerusalem
{roys02|omria01|arir}@cs.huji.ac.il

[2]Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
roiri@csail.mit.edu

Abstract

Dependency parsing is a central NLP task. In this paper we show that the common evaluation for unsupervised dependency parsing is highly sensitive to problematic annotations. We show that for three leading unsupervised parsers (Klein and Manning, 2004; Cohen and Smith, 2009; Spitkovsky et al., 2010a), a small set of parameters can be found whose modification yields a significant improvement in standard evaluation measures. These parameters correspond to local cases where no linguistic consensus exists as to the proper gold annotation. Therefore, the standard evaluation does not provide a true indication of algorithm quality. We present a new measure, *Neutral Edge Direction* (NED), and show that it greatly reduces this undesired phenomenon.

1 Introduction

Unsupervised induction of dependency parsers is a major NLP task that attracts a substantial amount of research (Klein and Manning, 2004; Cohen et al., 2008; Headden et al., 2009; Spitkovsky et al., 2010a; Gillenwater et al., 2010; Berg-Kirkpatrick et al., 2010; Blunsom and Cohn, 2010, *inter alia*). Parser quality is usually evaluated by comparing its output to a gold standard whose annotations are linguistically motivated. However, there are cases in which there is no linguistic consensus as to what the correct annotation is (Kübler et al., 2009). Examples include which verb is the head in a verb group structure (e.g., “can” or “eat” in “can eat”), and which noun is the head in a sequence of proper nouns (e.g., “John” or “Doe” in “John Doe”). We refer to such annotations as (*linguistically*) *problematic*. For such cases, evaluation measures should not punish the algorithm for deviating from the gold standard.

In this paper we show that the evaluation measures reported in current works are highly sensitive to the annotation in problematic cases, and propose a simple new measure that greatly neutralizes the problem.

We start from the following observation: for three leading algorithms (Klein and Manning, 2004; Cohen and Smith, 2009; Spitkovsky et al., 2010a), a small set (at most 18 out of a few thousands) of parameters can be found whose modification dramatically improves the standard evaluation measures (the attachment score measure by 9.3-15.1%, and the undirected measure by a smaller but still significant 1.3-7.7%). The phenomenon is implementation independent, occurring with several algorithms based on a fundamental probabilistic dependency model[1].

We show that these parameter changes can be mapped to edge direction changes in local structures in the dependency graph, and that these correspond to problematic annotations. Thus, the standard evaluation measures do not reflect the true quality of the evaluated algorithm.

We explain why the standard undirected evaluation measure is in fact sensitive to such edge direc-

[*] Omri Abend is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship.

[1] It is also language-independent; we have produced it in five different languages: English, Czech, Japanese, Portuguese, and Turkish. Due to space considerations, in this paper we focus on English, because it is the most studied language for this task and the most practically useful one at present.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 663–672,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

tion changes, and present a new evaluation measure, *Neutral Edge Direction* (NED), which greatly alleviates the problem by ignoring the edge direction in local structures. Using NED, manual modifications of model parameters always yields small performance differences. Moreover, NED sometimes punishes such manual parameter tweaking by yielding worse results. We explain this behavior using an experiment revealing that NED always prefers the structures that are more consistent with the modeling assumptions lying in the basis of the algorithm. When manual parameter modification is done against this preference, the NED results decrease.

The contributions of this paper are as follows. First, we show the impact of a small number of annotation decisions on the performance of unsupervised dependency parsers. Second, we observe that often these decisions are linguistically controversial and therefore this impact is misleading. This reveals a problem in the common evaluation of unsupervised dependency parsing. This is further demonstrated by noting that recent papers evaluate the task using three gold standards which differ in such decisions and which yield substantially different results. Third, we present the NED measure, which is agnostic to errors arising from choosing the non-gold direction in such cases.

Section 2 reviews related work. Section 3 describes the performed parameter modifications. Section 4 discusses the linguistic controversies in annotating problematic dependency structures. Section 5 presents NED. Section 6 describes experiments with it. A discussion is given in Section 7.

2 Related Work

Grammar induction received considerable attention over the years (see (Clark, 2001; Klein, 2005) for reviews). For unsupervised dependency parsing, the *Dependency Model with Valence (DMV)* (Klein and Manning, 2004) was the first to beat the simple right-branching baseline. A technical description of DMV is given at the end of this section.

The great majority of recent works, including those experimented with in this paper, are elaborations of DMV. Smith and Eisner (2005) improved the DMV results by generalizing the function maximized by DMV's EM training algorithm. Smith and Eisner (2006) used a structural locality bias, experimenting on five languages. Cohen et al. (2008) extended DMV by using a variational EM training algorithm and adding logistic normal priors. Cohen and Smith (2009, 2010) further extended it by using a *shared* logistic normal prior which provided a new way to encode the knowledge that some POS tags are more similar than others. A bilingual joint learning further improved their performance.

Headden et al. (2009) obtained the best reported results on WSJ10 by using a lexical extension of DMV. Gillenwater et al. (2010) used posterior regularization to bias the training towards a small number of parent-child combinations. Berg-Kirkpatrick et al. (2010) added new features to the M step of the DMV EM procedure. Berg-Kirkpatrick and Klein (2010) used a phylogenetic tree to model parameter drift between different languages. Spitkovsky et al. (2010a) explored several training protocols for DMV. Spitkovsky et al. (2010c) showed the benefits of Viterbi ("hard") EM to DMV training. Spitkovsky et al. (2010b) presented a novel *lightly-supervised* approach that used hyper-text mark-up annotation of web-pages to train DMV.

A few non-DMV-based works were recently presented. Daumé III (2009) used shift-reduce techniques. Blunsom and Cohn (2010) used tree substitution grammar to achieve best results on WSJ$^{\infty}$.

Druck et al. (2009) took a semi-supervised approach, using a set of rules such as "A noun is usually the parent of a determiner which is to its left", experimenting on several languages. Naseem et al. (2010) further extended this idea by using a single set of rules which globally applies to six different languages. The latter used a model similar to DMV.

The controversial nature of some dependency structures was discussed in (Nivre, 2006; Kübler et al., 2009). Klein (2005) discussed controversial constituency structures and the evaluation problems stemming from them, stressing the importance of a consistent standard of evaluation.

A few works explored the effects of annotation conventions on parsing performance. Nilsson et al. (2006) transformed the dependency annotations of coordinations and verb groups in the Prague TreeBank. They trained the supervised MaltParser (Nivre et al., 2006) on the transformed data, parsed the test data and re-transformed the resulting parse,

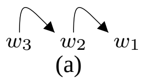
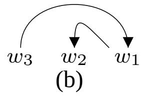

Figure 1: A dependency structure on the words w_1, w_2, w_3 before (Figure 1(a)) and after (Figure 1(b)) an *edge-flip* of $w_2 \rightarrow w_1$.

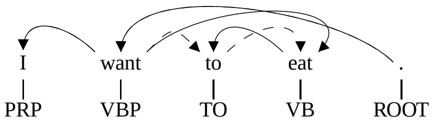

Figure 2: A parse of the sentence "I want to eat", before (straight line) and after (dashed line) an *edge-flip* of the edge "to"←"eat".

thus improving performance. Klein and Manning (2004) observed that a large portion of their errors is caused by predicting the wrong direction of the edge between a noun and its determiner. Kübler (2005) compared two different conversion schemes in German supervised constituency parsing and found one to have positive influence on parsing quality.

Dependency Model with Valence (DMV). DMV (Klein and Manning, 2004) defines a probabilistic grammar for unlabeled dependency structures. It is defined as follows: the root of the sentence is first generated, and then each head recursively generates its right and left dependents. The parameters of the model are of two types: P_{STOP} and P_{ATTACH}. $P_{STOP}(dir, h, adj)$ determines the probability to stop generating arguments, and is conditioned on 3 arguments: the head h, the direction dir ($(L)eft$ or $(R)ight$) and adjacency adj (whether the head already has dependents ($(Y)es$) in direction dir or not ($(N)o$)). $P_{ATTACH}(arg|h, dir)$ determines the probability to generate arg as head h's dependent in direction dir.

3 Significant Effects of Edge Flipping

In this section we present recurring error patterns in some of the leading unsupervised dependency parsers. These patterns are all local, confined to a sequence of up to three words (but mainly of just two consecutive words). They can often be mended by changing the directions of a few types of edges.

The modified parameters described in this section were handpicked to improve performance: we examined the local parser errors occurring the largest number of times, and found the corresponding parameters. Note that this is a valid methodology, since our goal is not to design a new algorithm but to demonstrate that modifying a small set of parameters can yield a major performance boost and eventually discover problems with evaluation methods or algorithms.

We start with a few definitions. Consider Figure 1(a) that shows a dependency structure on the words w_1, w_2, w_3. Edge flipping (henceforth, *edge-flip*) the edge $w_2 \rightarrow w_1$ is the following modification of a parse tree: (1) setting w_2's parent as w_1 (instead of the other way around), and (2) setting w_1's parent as w_3 (instead of the edge $w_3 \rightarrow w_2$). Figure 1(b) shows the dependency structure after the *edge-flip*.

Note that (1) imposes setting a new parent to w_2, as otherwise it would have had no parent. Setting this parent to be w_3 is the minimal modification of the original parse, since it does not change the attachment of the structure $[w_2, w_1]$ to the rest of the sentence, but only the direction of the internal edge.

Figure 2 presents a parse of the sentence "I want to eat", before and after an *edge-flip* of the edge "to"←"eat".

Since unsupervised dependency parsers are generally structure prediction models, the predictions of the parse edges are not independent. Therefore, there is no single parameter which completely controls the edge direction, and hence there is no direct way to perform an *edge-flip* by parameter modification. However, setting extreme values for the parameters controlling the direction of a certain edge type creates a strong preference towards one of the directions, and effectively determines the edge direction. This procedure is henceforth termed *parameter-flip*.

We show that by performing a few *parameter-flips*, a substantial improvement in the attachment score can be obtained. Results are reported for three algorithms.

Parameter Changes. All the works experimented with in this paper are not lexical and use sequences of POS tags as their input. In addition, they all use the DMV parameter set (P_{STOP} and P_{ATTACH}) for parsing. We will henceforth refer to this set, conditioned on POS tags, as the model parameter set.

We show how an edge in the dependency graph is encoded using the DMV parameters. Say the model prefers setting “to” (POS tag: TO) as a dependent of the infinitive verb (POS tag: VB) to its right (e.g., “to eat”). This is reflected by a high value of $P_{ATTACH}(TO|VB, L)$, a low value of $P_{ATTACH}(VB|TO, R)$, since “to” tends to be a left dependent of the verb and not the other way around, and a low value of $P_{STOP}(VB, L, N)$, as the verb usually has at least one left argument (i.e., “to”).

A *parameter-flip* of $w_1 \rightarrow w_2$ is hence performed by setting $P_{ATTACH}(w_2|w_1, R)$ to a very low value and $P_{ATTACH}(w_1|w_2, L)$ to a very high value. When the modifications to P_{ATTACH} are insufficient to modify the edge direction, $P_{STOP}(w_2, L, N)$ is set to a very low value and $P_{STOP}(w_1, R, N)$ to a very high value[2].

Table 1 describes the changes made for the three algorithms. The ‘$+$’ signs in the table correspond to edges in which the algorithm disagreed with the gold standard, and were thus modified. Similarly, the ‘–’ signs in the table correspond to edges in which the algorithm agreed with the gold standard, and were thus not modified. The number of modified parameters does not exceed 18 (out of a few thousands).

The *Freq*. column in the table shows the percentage of the tokens in sections 2-21 of PTB WSJ that participate in each structure. Equivalently, the percentage of edges in the corpus which are of either of the types appearing in the *Orig*. *Edge* column. As the table shows, the modified structures cover a significant portion of the tokens. Indeed, 42.9% of the tokens in the corpus participate in at least one of them[3].

Experimenting with Edge Flipping. We experimented with three DMV-based algorithms: a replication of (Klein and Manning, 2004), as appears in (Cohen et al., 2008) (henceforth, *km04*), Cohen and Smith (2009) (henceforth, *cs09*), and Spitkovsky et al. (2010a) (henceforth, *saj10a*). Decoding is done using the Viterbi algorithm[4]. For each of these algorithms we present the performance gain when compared to the original parameters.

The training set is sections 2-21 of the Wall Street Journal Penn TreeBank (Marcus et al., 1993). Testing is done on section 23. The constituency annotation was converted to dependencies using the rules of (Yamada and Matsumoto, 2003)[5].

Structure	Freq.	Orig. Edge	*km04*	*cs09*	*saj10a*
Coordination (“John & Mary”)	2.9%	$CC \rightarrow NNP$	−	+	−
Prepositional Phrase (“in the house”)	32.7%	$DT \rightarrow NN$	+	+	+
		$DT \rightarrow NNP$	−	+	+
		$DT \rightarrow NNS$	−	−	+
		$IN \rightarrow DT$	+	+	−
		$IN \leftarrow NN$	+	+	−
		$IN \leftarrow NNP$	+	−	−
		$IN \leftarrow NNS$	−	+	−
		$PRP\$ \rightarrow NN$	−	−	+
Modal Verb (“can eat”)	2.4%	$MD \leftarrow VB$	−	+	−
Infinitive Verb (“to eat”)	4.5%	$TO \rightarrow VB$	−	+	+
Proper Name Sequence (“John Doe”)	18.5%	$NNP \rightarrow NNP$	+	−	−

Table 1: Parameter changes for the three algorithms. The Freq. column shows what percentage of the tokens in sections 2-21 of PTB WSJ participate in each structure. The Orig. column indicates the original edge. The modified edge is of the opposite direction. The other columns show the different algorithms: *km04*: basic DMV model (replication of (Klein and Manning, 2004)); *cs09*; (Cohen and Smith, 2009); *saj10a*: (Spitkovsky et al., 2010a).

Following standard practice, we present the attachment score (i.e., percentage of words that have a correct head) of each algorithm, with both the original parameters and the modified ones. We present results both on all sentences and on sentences of length ≤ 10, excluding punctuation.

Table 2 shows results for all algorithms[6]. The performance difference between the original and the modified parameter set is considerable for all data sets, where differences exceed 9.3%, and go up to 15.1%. These are enormous differences from the perspective of current algorithm evaluation results.

4 Linguistically Problematic Annotations

In this section, we discuss the controversial nature of the annotation in the modified structures (Kübler

[2] Note that this yields unnormalized models. Again, this is justified since the resulting model is only used as a basis for discussion and is not a fully fledged algorithm.

[3] Some tokens participate in more than one structure.

[4] http://www.cs.cmu.edu/∼scohen/parser.html.

[5] http://www.jaist.ac.jp/∼h-yamada/

[6] Results are slightly worse than the ones published in the original papers due to the different decoding algorithms (*cs09* use MBR while we used Viterbi) and a different conversion procedure (*saj10a* used (Collins, 1999) and not (Yamada and Matsumoto, 2003)) ; see Section 5.

Algo.	≤ 10			$\leq \infty$		
	$Orig.$	$Mod.$	Δ	$Orig.$	$Mod.$	Δ
km04	45.8	59.8	14	34.6	43.9	9.3
cs09	60.9	72.9	12	39.9	54.6	14.7
saj10a	54.7	69.8	15.1	41.6	54.3	12.7

Table 2: Results of the original ($Orig.$ columns), the modified ($Mod.$ columns) parameter sets and their difference (Δ columns) for the three algorithms.

et al., 2009). We remind the reader that structures for which no linguistic consensus exists as to their correct annotation are referred to as (linguistically) problematic.

We begin by showing that all the structures modified are indeed linguistically problematic. We then note that these controversies are reflected in the evaluation of this task, resulting in three, significantly different, gold standards currently in use.

Coordination Structures are composed of two proper nouns, separated by a conjunctor (e.g., "John and Mary"). It is not clear which token should be the head of this structure, if any (Nilsson et al., 2006).

Prepositional Phrases (e.g., "in the house" or "in Rome"), where every word is a reasonable candidate to head this structure. For example, in the annotation scheme used by (Collins, 1999) the preposition is the head, in the scheme used by (Johansson and Nugues, 2007) the noun is the head, while TUT annotation, presented in (Bosco and Lombardo, 2004), takes the determiner to be the noun's head.

Verb Groups are composed of a verb and an auxiliary or a modal verb (e.g., "can eat"). Some schemes choose the modal as the head (Collins, 1999), others choose the verb (Rambow et al., 2002).

Infinitive Verbs (e.g., "to eat") are also in controversy, as in (Yamada and Matsumoto, 2003) the verb is the head while in (Collins, 1999; Bosco and Lombardo, 2004) the "to" token is the head.

Sequences of Proper Nouns (e.g., "John Doe") are also subject to debate, as PTB's scheme takes the last proper noun as the head, and BIO's scheme defines a more complex scheme (Dredze et al., 2007).

Evaluation Inconsistency Across Papers. A fact that may not be recognized by some readers is that comparing the results of unsupervised dependency parsers across different papers is not directly possible, since different papers use different gold standard annotations *even when they are all derived from the Penn Treebank constituency annotation*. This happens because they use different rules for converting constituency annotation to dependency annotation. A probable explanation for this fact is that people have tried to correct linguistically problematic annotations in different ways, which is why we note this issue here[7].

There are three different annotation schemes in current use: (1) Collins head rules (Collins, 1999), used in e.g., (Berg-Kirkpatrick et al., 2010; Spitkovsky et al., 2010a); (2) Conversion rules of (Yamada and Matsumoto, 2003), used in e.g., (Cohen and Smith, 2009; Gillenwater et al., 2010); (3) Conversion rules of (Johansson and Nugues, 2007) used, e.g., in the CoNLL shared task 2007 (Nivre et al., 2007) and in (Blunsom and Cohn, 2010).

The differences between the schemes are substantial. For instance, 14.4% of section 23 is tagged differently by (1) and (2)[8].

5 The Neutral Edge Direction (NED) Measure

As shown in the previous sections, the annotation of problematic edges can substantially affect performance. This was briefly discussed in (Klein and Manning, 2004), which used undirected evaluation as a measure which is less sensitive to alternative annotations. Undirected accuracy was commonly used since to assess the performance of unsupervised parsers (e.g., (Smith and Eisner, 2006; Headden et al., 2008; Spitkovsky et al., 2010a)) but also of supervised ones (Wang et al., 2005; Wang et al., 2006). In this section we discuss why this measure is in fact not indifferent to *edge-flips* and propose a new measure, Neutral Edge Direction (NED).

[7] Indeed, half a dozen flags in the LTH Constituent-to-Dependency Conversion Tool (Johansson and Nugues, 2007) are used to control the conversion in problematic cases.

[8] In our experiments we used the scheme of (Yamada and Matsumoto, 2003), see Section 3. The significant effects of edge flipping were observed with the other two schemes as well.

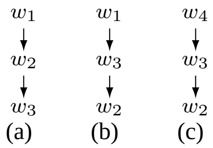

Figure 3: A dependency structure on the words w_1, w_2, w_3 before (Figure 3(a)) and after (Figure 3(b)) an *edge-flip* of $w_2 \rightarrow w_3$, and when the direction of the edge between w_2 and w_3 is switched and the new parent of w_3 is set to be some other word, w_4 (Figure 3(c)).

Undirected Evaluation. The measure is defined as follows: traverse over the tokens and mark a correct attachment if the token's induced parent is either (1) its gold parent or (2) its gold child. The score is the ratio of correct attachments and the number of tokens.

We show that this measure does not ignore *edge-flips*. Consider Figure 3 that shows a dependency structure on the words w_1, w_2, w_3 before (Figure 3(a)) and after (Figure 3(b)) an *edge-flip* of $w_2 \rightarrow w_3$. Assume that 3(a) is the gold standard and that 3(b) is the induced parse. Consider w_2. Its induced parent (w_3) is its gold child, and thus undirected evaluation does not consider it an error. On the other hand, w_3 is assigned w_2's gold parent, w_1. This is considered an error, since w_1 is neither w_3's gold parent (as it is w_2), nor its gold child[9]. Therefore, one of the two tokens involved in the *edge-flip* is penalized by the measure.

Recall the example "I want to eat" and the *edge-flip* of the edge "to"←"eat" (Figure 2). As "to"'s parent in the induced graph ("want") is neither its gold parent nor its gold child, the undirected evaluation measure marks it as an error. This is an example where an *edge-flip* in a problematic edge, which should not be considered an error, was in fact considered an error by undirected evaluation.

Neutral Edge Direction (NED). The NED measure is a simple extension of the undirected evaluation measure[10]. Unlike undirected evaluation, NED ignores all errors directly resulting from an *edge-flip*.

NED is defined as follows: traverse over the tokens and mark a correct attachment if the token's induced parent is either (1) its gold parent (2) its gold child or (3) its gold grandparent. The score is the ratio of correct attachments and the number of tokens.

NED, by its definition, ignores *edge-flips*. Consider again Figure 3, where we assume that 3(a) is the gold standard and that 3(b) is the induced parse. Much like undirected evaluation, NED will mark the attachment of w_2 as correct, since its induced parent is its gold child. However, unlike undirected evaluation, w_3's induced attachment will also be marked as correct, as its induced parent is its gold grandparent.

Now consider another induced parse in which the direction of the edge between w_2 and w_3 is switched and the w_3's parent is set to be some other word, w_4 (Figure 3(c)). This should be marked as an error, even if the direction of the edge between w_2 and w_3 is controversial, since the structure $[w_2, w_3]$ is no longer a dependent of w_1. It is indeed a NED error. Note that undirected evaluation gives the parses in Figure 3(b) and Figure 3(c) the same score, while if the structure $[w_2, w_3]$ is problematic, there is a major difference in their correctness.

Discussion. Problematic structures are ubiquitous, with more than 40% of the tokens in PTB WSJ appearing in at least one of them (see Section 3). Therefore, even a substantial difference in the attachment between two parsers is not necessarily indicative of a true quality difference. However, an attachment score difference that persists under NED is an indication of a true quality difference, since generally problematic structures are local (i.e., obtained by an *edge-flip*) and NED ignores such errors.

Reporting NED alone is insufficient, as obviously the edge direction does matter in some cases. For example, in adjective–noun structures (e.g., "big house"), the correct edge direction is widely agreed upon ("big"←"house") (Kübler et al., 2009), and thus choosing the wrong direction should be considered an error. Therefore, we suggest evaluating using both NED and attachment score in order to get a full picture of the parser's performance.

A possible criticism on NED is that it is only indifferent to alternative annotations in structures of size 2 (e.g., "to eat") and does not necessarily handle larger problematic structures, such as coordinations

[9] Otherwise, the gold parse would have contained a $w_1 \rightarrow w_2 \rightarrow w_3 \rightarrow w_1$ cycle.

[10] An implementation of NED is available at http://www.cs.huji.ac.il/∼roys02/software/ned.html

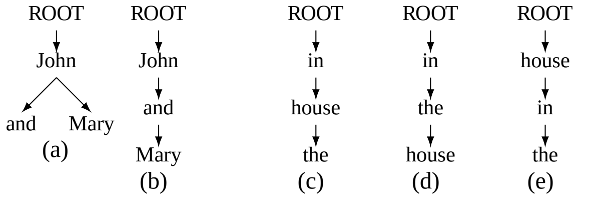

Figure 4: Alternative parses of "John and Mary" and "in the house". Figure 4(a) follows (Collins, 1999), Figure 4(b) follows (Johansson and Nugues, 2007). Figure 4(c) follows (Collins, 1999; Yamada and Matsumoto, 2003). Figure 4(d) and Figure 4(e) show induced parses made by (*km04*,*saj10a*) and *cs09*, respectively.

(see Section 4). For example, Figure 4(a) and Figure 4(b) present two alternative annotations of the sentence "John and Mary". Assume the parse in Figure 4(a) is the gold parse and that in Figure 4(b) is the induced parse. The word "Mary" is a NED error, since its induced parent ("and") is neither its gold child nor its gold grandparent. Thus, NED does not accept all possible annotations of structures of size 3. On the other hand, using a method which accepts all possible annotations of structures of size 3 seems too permissive. A better solution may be to modify the gold standard annotation, so to explicitly annotate problematic structures as such. We defer this line of research to future work.

NED is therefore an evaluation measure which is indifferent to *edge-flips*, and is consequently less sensitive to alternative annotations. We now show that NED is indifferent to the differences between the structures originally learned by the algorithms mentioned in Section 3 and the gold standard annotation in all the problematic cases we consider.

Most of the modifications made are *edge-flips*, and are therefore ignored by NED. The exceptions are coordinations and prepositional phrases which are structures of size 3. In the former, the alternative annotations differ only in a single *edge-flip* (i.e., $CC \rightarrow NNP$), and are thus not NED errors. Regarding prepositional phrases, Figure 4(c) presents the gold standard of "in the house", Figure 4(d) the parse induced by *km04* and *saj10a* and Figure 4(e) the parse induced by *cs09*. As the reader can verify, both induced parses receive a perfect NED score.

In order to further demonstrate NED's insensitivity to alternative annotations, we took two of the three common gold standard annotations (see Section 4) and evaluated them one against the other. We considered section 23 of WSJ following the scheme of (Yamada and Matsumoto, 2003) as the gold standard and of (Collins, 1999) as the evaluated set. Results show that the attachment score is only 85.6%, the undirected accuracy is improved to 90.3%, while the NED score is 95.3%. This shows that NED is significantly less sensitive to the differences between the different annotation schemes, compared to the other evaluation measures.

6 Experimenting with NED

In this section we show that NED indeed reduces the performance difference between the original and the modified parameter sets, thus providing empirical evidence for its validity. For brevity, we present results only for the entire WSJ corpus. Results on WSJ10 are similar. The datasets and decoding algorithms are the same as those used in Section 3.

Table 3 shows the score differences between the parameter sets using attachment score, undirected evaluation and NED. A substantial difference persists under undirected evaluation: a gap of 7.7% in *cs09*, of 3.5% in *saj10a* and of 1.3% in *km04*.

The differences are further reduced using NED. This is consistent with our discussion in Section 5, and shows that undirected evaluation only ignores some of the errors inflicted by *edge-flips*.

For *cs09*, the difference is substantially reduced, but a 4.2% performance gap remains. For *km04* and *saj10a*, the original parameters outperform the new ones by 3.6% and 1% respectively.

We can see that even when ignoring *edge-flips*, some difference remains, albeit not necessarily in the favor of the modified models. This is because we did not directly perform *edge-flips*, but rather *parameter-flips*. The difference is thus a result of second-order effects stemming from the *parameter-flips*. In the next section, we explain why the remaining difference is positive for some algorithms (*cs09*) and negative for others (*km04*, *saj10a*).

For completeness, Table 4 shows a comparison of some of the current state-of-the-art algorithms, using attachment score, undirected evaluation and NED. The training and test sets are those used in Section 3. The table shows that the relative orderings of the algorithms under NED is different than under the other

Algo.	$Mod. - Orig.$		
	Attach.	Undir.	NED
km04	9.3 (43.9–34.6)	1.3 (54.2–52.9)	–3.6 (63–66.6)
cs09	14.7 (54.6–39.9)	7.7 (56.9–49.2)	4.2 (66.8–62.6)
saj10a	12.7 (54.3–41.6)	3.5 (59.4–55.9)	–1 (66.8–67.8)

Table 3: Differences between the modified and original parameter sets when evaluated using attachment score (*Attach.*), undirected evaluation (*Undir.*), and NED.

measures. This is an indication that NED provides a different perspective on algorithm quality[11].

Algo.	Att_{10}	Att_{∞}	Un_{10}	Un_{∞}	NED$_{10}$	NED$_{\infty}$
bbdk10	66.1	49.6	70.1	56.0	75.5	61.8
bc10	**67.2**	**53.6**	**73**	**61.7**	81.6	70.2
cs09	61.5	42	66.9	50.4	81.5	62.9
gggtp10	57.1	45	62.5	53.2	80.4	65.1
km04	45.8	34.6	60.3	52.9	78.4	66.6
saj10a	54.7	41.6	66.5	55.9	78.9	67.8
saj10c	63.8	46.1	72.6	58.8	**84.2**	**70.8**
*saj10b**	67.9	48.2	74.0	57.7	86.0	70.7

Table 4: A comparison of recent works, using Att (*attachment score*) Un (*undirected evaluation*) and NED, on sentences of length $\leq$ 10 (excluding punctuation) and on all sentences. The gold standard is obtained using the rules of (Yamada and Matsumoto, 2003). *bbdk10*: (Berg-Kirkpatrick et al., 2010), *bc10*: (Blunsom and Cohn, 2010), *cs09*: (Cohen and Smith, 2009), *gggtp10*: (Gillenwater et al., 2010), *km04*: A replication of (Klein and Manning, 2004), *saj10a*: (Spitkovsky et al., 2010a), *saj10c*: (Spitkovsky et al., 2010c), *saj10b**: A lightly-supervised algorithm (Spitkovsky et al., 2010b).

7 Discussion

In this paper we explored two ways of dealing with cases in which there is no clear theoretical justification to prefer one dependency structure over another. Our experiments suggest that it is crucial to deal with such structures if we would like to have a proper evaluation of unsupervised parsing algorithms against a gold standard.

The first way was to modify the parameters of the parsing algorithms so that in cases where such problematic decisions are to be made they follow the gold standard annotation. Indeed, this modification leads to a substantial improvement in the attachment score of the algorithms.

The second way was to change the evaluation. The NED measure we proposed does not punish for differences between gold and induced structures in the problematic cases. Indeed, in Section 6 (Table 3) we show that the differences between the original and modified models are much smaller when evaluating with NED compared to when evaluating with the traditional attachment score.

As Table 3 reveals, however, even when evaluating with NED, there is still some difference between the original and the modified model, for each of the algorithms we consider. Moreover, for two of the algorithms (*km04* and *saj10a*) NED prefers the original model while for one (*cs09*) it prefers the modified version. In this section we explain these patterns and show that they are both consistent and predictable.

Our hypothesis, for which we provide empirical justification, is that in cases where there is no theoretically preferred annotation, NED prefers the structures that are more learnable by DMV. That is, NED gives higher scores to the annotations that better fit the assumptions and modeling decisions of DMV, the model that lies in the basis of the parsing algorithms.

To support our hypothesis we perform an experiment requiring two preparatory steps for each algorithm. First, we construct a supervised version of the algorithm. This supervised version consists of the same statistical model as the original unsupervised algorithm, but the parameters are estimated to maximize the likelihood of a *syntactically annotated* training corpus, rather than of a plain text corpus.

Second, we construct two corpora for the algorithm, both consist of the same text and differ only in their syntactic annotation. The first is annotated with the gold standard annotation. The second is similarly annotated except in the linguistically problematic structures. We replace these structures with the ones that would have been created with the unsupervised version of the algorithm (see Table 1 for the relevant structures for each algorithm)[12]. Each

[11] Results may be different than the ones published in the original papers due to the different conversion procedures used in each work. See Section 4 for discussion.

[12] In cases the structures are comprised of a single edge, the second corpus is obtained from the gold standard by an *edge-flip*. The only exceptions are the cases of the prepositional phrases. Their gold standard and the learned structures for each of the algorithms are shown in Figure 4. In this case, the second corpus is obtained from the gold standard by replacing each prepositional phrase in the gold standard with the corresponding

corpus is divided into a training and a test set.

We then train the supervised version of the algorithms on each of the training sets. We parse the test data twice, once with each of the resulting models. We evaluate both parsed corpora against the corpus annotation from which they originated.

The training set of each corpus consists of sections 2–21 of WSJ20 (i.e., WSJ sentences of length $\leq$20, excluding punctuation)[13] and the test set is section 23 of WSJ$^\infty$. Evaluation is performed using both NED and attachment score. The patterns we observed are very similar for both. For brevity, we report only attachment score results.

	km04		*cs09*		*saj10a*	
	Orig.	*Gold*	*Orig.*	*Gold*	*Orig.*	*Gold*
NED, Unsup.	**66.6**	63	62.6	**66.8**	**67.8**	66.8
Sup.	**71.3**	69.9	63.3	**69.9**	**71.8**	69.9

Table 5: The first line shows the NED results from Section 6, when using the original parameters (*Orig.* columns) and the modified parameters (*Gold* columns). The second line shows the results of the supervised versions of the algorithms using the corpus which agrees with the unsupervised model in the problematic cases (*Orig.*) and the gold standard (*Gold*).

The results of our experiment are presented in Table 5 along with a comparison to the NED scores from Section 6. The table clearly demonstrates that a set of parameters (original or modified) is preferred by NED in the unsupervised experiments reported in Section 6 (top line) if and only if the structures produced by this set are better learned by the supervised version of the algorithm (bottom line).

This observation supports our hypothesis that in cases where there is no theoretical preference for one structure over the other, NED (unlike the other measures) prefers the structures that are more consistent with the modeling assumptions lying in the basis of the algorithm. We consider this to be a desired property of a measure since a more consistent model should be preferred where no theoretical preference exists.

8 Conclusion

In this paper we showed that the standard evaluation of unsupervised dependency parsers is highly sensitive to problematic annotations. We modified a small set of parameters that controls the annotation in such problematic cases in three leading parsers. This resulted in a major performance boost, which is unindicative of a true difference in quality.

We presented *Neutral Edge Direction* (NED), a measure that is less sensitive to the annotation of local structures. As the problematic structures are generally local, NED is less sensitive to their alternative annotations. In the future, we suggest reporting NED along with the current measures.

Acknowledgements. We would like to thank Shay Cohen for his assistance with his implementation of the DMV parser and Taylor Berg-Kirkpatrick, Phil Blunsom and Jennifer Gillenwater for providing us with their data sets. We would also like to thank Valentin I. Spitkovsky for his comments and for providing us with his data sets.

References

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero and Dan Klein, 2010. *Painless unsupervised learning with features*. In *Proc. of NAACL*.

Taylor Berg-Kirkpatrick and Dan Klein, 2010. *Phylogenetic Grammar Induction*. In *Proc. of ACL*.

Cristina Bosco and Vincenzo Lombardo, 2004. *Dependency and relational structure in treebank annotation*. In *Proc. of the Workshop on Recent Advances in Dependency Grammar at COLING'04*.

Phil Blunsom and Trevor Cohn, 2010. *Unsupervised Induction of Tree Substitution Grammars for Dependency Parsing*. In *Proc. of EMNLP*.

Shay B. Cohen, Kevin Gimpel and Noah A. Smith, 2008. *Logistic Normal Priors for Unsupervised Probabilistic Grammar Induction*. In *Proc. of NIPS*.

Shay B. Cohen and Noah A. Smith, 2009. *Shared Logistic Normal Distributions for Soft Parameter Tying*. In *Proc. of HLT-NAACL*.

Michael J. Collins, 1999. *Head-driven statistical models for natural language parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia.

Alexander Clark, 2001. *Unsupervised language acquisition: theory and practice*. Ph.D. thesis, University of Sussex.

Hal Daumé III, 2009. *Unsupervised search-based structured prediction*. In *Proc. of ICML*.

learned structure.

[13] In using WSJ20, we follow (Spitkovsky et al., 2010a), which showed that training the DMV on sentences of bounded length yields a higher score than using the entire corpus. We use it as we aim to use an optimal setting.

Mark Dredze, John Blitzer, Partha Pratim Talukdar, Kuzman Ganchev, João V. Graça and Fernando Pereira, 2007. *Frustratingly Hard Domain Adaptation for Dependency Parsing*. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

Gregory Druck, Gideon Mann and Andrew McCallum, 2009. *Semi-supervised learning of dependency parsers using generalized expectation criteria*. In *Proc. of ACL*.

Jennifer Gillenwater, Kuzman Ganchev, João V. Graça, Ben Taskar and Fernando Preira, 2010. *Sparsity in dependency grammar induction*. In *Proc. of ACL*.

William P. Headden III, David McClosky, and Eugene Charniak, 2008. *Evaluating Unsupervised Part-of-Speech Tagging for Grammar Induction*. In *Proc. of COLING*.

William P. Headden III, Mark Johnson and David McClosky, 2009. *Improving unsupervised dependency parsing with richer contexts and smoothing*. In *Proc. of HLT-NAACL*.

Richard Johansson and Pierre Nugues, 2007. *Extended Constituent-to-Dependency Conversion for English*. In *Proc. of NODALIDA*.

Dan Klein, 2005. *The unsupervised learning of natural language structure*. Ph.D. thesis, Stanford University.

Dan Klein and Christopher Manning, 2004. *Corpus-based induction of syntactic structure: Models of dependency and constituency*. In *Proc. of ACL*.

Sandra Kübler, 2005. *How Do Treebank Annotation Schemes Influence Parsing Results? Or How Not to Compare Apples And Oranges*. In *Proc. of RANLP*.

Sandra Kübler, R. McDonald and Joakim Nivre, 2009. *Dependency Parsing*. Morgan And Claypool Publishers.

Mitchell Marcus, Beatrice Santorini and Mary Ann Marcinkiewicz, 1993. *Building a large annotated corpus of English: The Penn treebank*. *Computational Linguistics* 19:313-330.

Tahira Naseem, Harr Chen, Regina Barzilay and Mark Johnson, 2010. *Using universal linguistic knowledge to guide grammar induction*. In *Proc. of EMNLP*.

Joakim Nivre, 2006. *Inductive Dependency Parsing*. Springer.

Joakim Nivre, Johan Hall and Jens Nilsson, 2006. *MaltParser: A data-driven parser-generator for dependency parsing*. In *Proc. of LREC-2006*.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel and Deniz Yuret, 2007. *The CoNLL 2007 shared task on dependency parsing*. In *Proc. of the CoNLL Shared Task, EMNLP-CoNLL, 2007*.

Jens Nilsson, Joakim Nivre and Johan Hall, 2006. *Graph transformations in data-driven dependency parsing*. In *Proc. of ACL*.

Owen Rambow, Cassandre Creswell, Rachel Szekely, Harriet Tauber and Marilyn Walker, 2002. *A dependency treebank for English*. In *Proc. of LREC*.

Noah A. Smith and Jason Eisner, 2005. *Guiding unsupervised grammar induction using contrastive estimation*. In *Proc. of IJCAI*.

Noah A. Smith and Jason Eisner, 2006. *Annealing structural bias in multilingual weighted grammar induction*. In *Proc. of ACL*.

Valentin I. Spitkovsky, Hiyan Alshawi and Daniel Jurafsky, 2010a. *From Baby Steps to Leapfrog: How "Less is More" in Unsupervised Dependency Parsing*. In *Proc. of NAACL-HLT*.

Valentin I. Spitkovsky, Hiyan Alshawi and Daniel Jurafsky, 2010b. *Profiting from Mark-Up: Hyper-Text Annotations for Guided Parsing*. In *Proc. of ACL*.

Valentin I. Spitkovsky, Hiyan Alshawi, Daniel Jurafsky and Christopher D. Manning, 2010c. *Viterbi training improves unsupervised dependency parsing*. In *Proc. of CoNLL*.

Qin Iris Wang, Dale Schuurmans and Dekang Lin, 2005. *Strictly Lexical Dependency Parsing*. In *IWPT*.

Qin Iris Wang, Colin Cherry, Dan Lizotte and Dale Schuurmans, 2006. *Improved Large Margin Dependency Parsing via Local Constraints and Laplacian Regularization*. In *Proc. of CoNLL*.

Hiroyasu Yamada and Yuji Matsumoto, 2003. *Statistical dependency analysis with support vector machines*. In *Proc. of the International Workshop on Parsing Technologies*.