Effects of Noun Phrase Bracketing in Dependency Parsing and Machine Translation

Nathan Green
Charles University in Prague
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
green@ufal.mff.cuni.cz

Abstract

Flat noun phrase structure was, up until recently, the standard in annotation for the Penn Treebanks. With the recent addition of internal noun phrase annotation, dependency parsing and applications down the NLP pipeline are likely affected. Some machine translation systems, such as TectoMT, use deep syntax as a language transfer layer. It is proposed that changes to the noun phrase dependency parse will have a cascading effect down the NLP pipeline and in the end, improve machine translation output, even with a reduction in parser accuracy that the noun phrase structure might cause. This paper examines this noun phrase structure's effect on dependency parsing, in English, with a maximum spanning tree parser and shows a 2.43%, 0.23 Bleu score, improvement for English to Czech machine translation.

1 Introduction

Noun phrase structure in the Penn Treebank has up until recently been only considered, due to underspecification, a flat structure. Due to the annotation and work of Vadas and Curran (2007a; 2007b; 2008), we are now able to create Natural Language Processing (NLP) systems that take advantage of the internal structure of noun phrases in the Penn Treebank. This extra internal structure introduces additional complications in NLP applications such as parsing.

Dependency parsing has been a prime focus of NLP research of late due to its ability to help parse languages with a free word order. Dependency parsing has been shown to improve NLP systems in certain languages and in many cases is considered the state of the art in the field. Dependency parsing made many improvements due to the CoNLL X shared task (Buchholz and Marsi, 2006). However, in most cases, these systems were trained with a flat noun phrase structure in the Penn Treebank. Vadas' internal noun phrase structure has been used in previous work on constituent parsing using Collin's parser (Vadas and Curran, 2007c), but has yet to be analyzed for its effects on dependency parsing.

Parsing is very early in the NLP pipeline. Therefore, improvements in parsing output could have an improvement on other areas of NLP in many cases, such as Machine Translation. At the same time, any errors in parsing will tend to propagate down the NLP pipeline. One would expect parsing accuracy to be reduced when the complexity of the parse is increased, such as adding noun phrase structure. But, for a machine translation system that is reliant on parsing, the new noun phrase structure, even with reduced parser accuracy, may yield improvements due to a more detailed grammatical structure. This is particularly of interest for dependency relations, as it may aid in finding the correct head of a term in a complex noun phrase.

This paper examines the results and errors in parsing and machine translation of dependency parsers, trained with annotated noun phrase structure, against those with a flat noun phrase structure. These results are compared with two systems: a Baseline Parser with no internally annotated noun phrases and a Gold NP Parser trained with data which contains

Proceedings of the ACL-HLT 2011 Student Session, pages 69–74,
Portland, OR, USA 19-24 June 2011.

gold standard internal noun phrase structure annotation. Additionally, we analyze the effect of these improvements and errors in parsing down the NLP pipeline on the TectoMT machine translation system (Žabokrtský et al., 2008).

Section 2 contains background information needed to understand the individual components of the experiments. The methodology used to carry out the experiments is described in Section 3. Results are shown and discussed in Section 4. Section 5 concludes and discusses future work and implications of this research.

2 Related Work

2.1 Dependency Parsing

Dependence parsing is an alternative view to the common phrase or constituent parsing techniques used with the Penn Treebank. Dependency relations can be used in many applications and have been shown to be quite useful in languages with a free word order. With the influx of many data-driven techniques, the need for annotated dependency relations is apparent. Since there are many data sets with constituent relations annotated, this paper uses free conversion software provided from the CoNLL 2008 shared task to create dependency relations (Johansson and Nugues, 2007; Surdeanu et al., 2008).

2.2 Dependency Parsers

Dependency parsing comes in two main forms: Graph algorithms and Greedy algorithms. The two most popular algorithms are McDonald's MSTParser (McDonald et al., 2005) and Nivre's MaltParser (Nivre, 2003). Each parser has its advantages and disadvantages, but the accuracy overall is approximately the same. The types of errors made by each parser, however, are very different. MSTParser is globally trained for an optimal solution and this has led it to get the best results on longer sentences. MaltParser on the other hand, is a greedy algorithm. This allows it to perform extremely well on shorter sentences, as the errors tend to propagate and cause more egregious errors in longer sentences with longer dependencies (McDonald and Nivre, 2007). We expect each parser to have different errors handling internal noun phrase structure, but for this paper we will only be examining the globally trained MSTParser.

2.3 TectoMT

TectoMT is a machine translation framework based on Praguian tectogrammatics (Sgall, 1967) which represents four main layers: word layer, morphological layer, analytical layer, and tectogrammatical layer (Popel et al., 2010). This framework is primarily focused on the translation from English into Czech. Since much of dependency parsing work has been focused on Czech, this choice of machine translation framework logically follows as TectoMT makes direct use of the dependency relationships. The work in this paper primarily addresses the noun phrase structure in the analytical layer (SEnglishA in Figure 1).

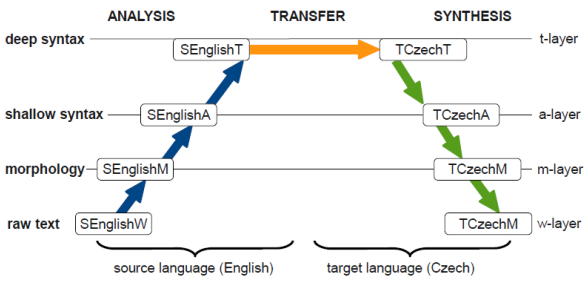

Figure 1: Translation Process in TectoMT in which the tectogrammatical layer is transfered from English to Czech.

TectoMT is a modular framework built in Perl. This allows great ease in adding the two different parsers into the framework since each experiment can be run as a separate "Scenario" comprised of different parsing "Blocks". This allows a simple comparison of two machine translation system in which everything remains constant except the dependency parser.

2.4 Noun Phrase Structure

The Penn Treebank is one of the most well known English language treebanks (Marcus et al., 1993), consisting of annotated portions of the Wall Street Journal. Much of the annotation task is painstakingly done by annotators in great detail. Some structures are not dealt with in detail, such as noun phrase structure. Not having this information makes it difficult to tell the dependencies on phrases such as

"crude oil prices" (Vadas and Curran, 2007c). Without internal annotation it is ambiguous whether the phrase is stating "crude prices" (crude (oil prices)) or "crude oil" ((crude oil) prices).

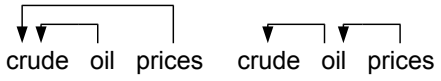

Figure 2: Ambiguous dependency caused by internal noun phrase structure.

Manual annotation of these phrases would be quite time consuming and as seen in the example above, sometimes ambiguous and therefore prone to poor inter-annotator agreement. Vadas and Curran have constructed a Gold standard version Penn treebank with these structures. They were also able to train supervised learners to an F-score of 91.44% (Vadas and Curran, 2007a; Vadas and Curran, 2007b; Vadas and Curran, 2008). The additional complexity of noun phrase structure has been shown to reduce parser accuracy in Collin's parser but no similar evaluation has been conducted for dependency parsers. The internal noun phrase structure has been used in experiments prior but without evaluation with respect to the noun phrases (Galley and Manning, 2009).

3 Methodology

The Noun Phrase Bracketing experiments consist of a comparison two systems.

1. The Baseline system is McDonald's MSTParser trained on the Penn Treebank in English without any extra noun phrase bracketing.

2. The Gold NP Parser is McDonald's MSTParser trained on the Penn Treebank in English with gold standard noun phrase structure annotations (Vadas and Curran, 2007a).

3.1 Data Sets

To maintain a consistent dataset to compare to previous work we use the Wall Street Journal (WSJ) section of the Penn Treebank since it was used in the CoNLL X shared task on dependency parsing (Buchholz and Marsi, 2006). Using the same common breakdown of datasets, we use WST section 02-21 for training and section 22 for testing, which allows us to have comparable results to previous works. To test the effects of the noun phrase structure on machine translation, ACL 2008's Workshop on Statistical Machine translation's (WMT) data are used.

3.2 Process Flow

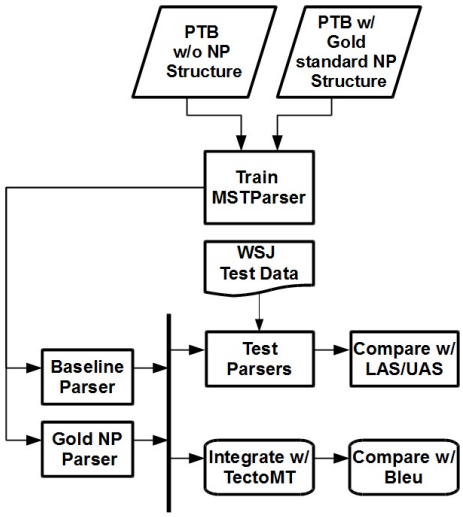

Figure 3: Experiment Process Flow. PTB (Penn Tree Bank), NP (Noun Phrase Structure), LAS (Labeled Accuracy Score), UAS (Unlabeled Accuracy Score), Wall Street Journal (WSJ)

We begin the the experiments by constructing two data sets:

1. The Penn Treebank with no internal noun phrase structure (PTB w/o NP structure).

2. The Penn Treebank with gold standard noun phrase annotations provided by Vadas and Curran (PTB w/ gold standard NP structure).

From these datasets we construct two separate parsers. These parsers are trained using McDonald's Maximum Spanning Tree Algorithm (MSTParser) (McDonald et al., 2005).

Both of the parsers are then tested on a subset of the WSJ corpus, section 22, of the Penn Treebank and the UAS and LAS scores are generated. Errors generated by each of these systems are then compared to discover where the internal noun phrase structure affects the output. Parser accuracy is not necessarily the most important aspect of this work.

The effect of this noun phrase structure down the NLP pipeline is also crucial. For this, the parsers are inserted into the TectoMT system.

3.3 Metrics

*L*abeled *A*ccuracy *S*core (LAS) and *U*nlabeled *A*ccuracy *S*core (UAS) are the primary ways to evaluate dependency parsers. UAS is the percentage of words that are correctly linked to their heads. LAS is the percentage of words that are connected to their correct heads and have the correct dependency label. UAS and LAS are used to compare one system against another, as was done in CoNLL X (Buchholz and Marsi, 2006).

The Bleu (*Bi*L*ingual* *E*valuation *U*nderstudy) score is an automatic scoring mechanism for machine translation that is quick and can be reused as a benchmark across machine translation tasks. Bleu is calculated as the geometric mean of n-grams comparing a machine translation and a reference text (Papineni et al., 2002). This experiment compares the two parsing systems against each other using the above metrics. In both cases the test set data is sampled 1,000 times without replacement to calculate statistical significance using a pairwise comparison.

4 Results and Discussion

When applied, the gold standard annotations changed approximately 1.5% of the edges in the training data. Once trained, both parsers were tested against section 22 of their respective annotated corpora. As Table 1 shows, the Baseline Parser obtained near identical LAS and UAS scores. This was expected given the additional complexity of predicting the noun phrase structure and the previous work on noun phrase bracketing's effect on Collin's parser.

Systems	LAS	UAS
Baseline Parser	88.12%	91.11%
Gold NP Parser	88.10%	91.10%

Table 1: Parsing results for the Baseline and Gold NP Parsers. Each is trained on Section 02-21 of the WSJ and tested on Section 22

While possibly more error prone, the 1.5% change in edges in the training data did appear to add more useful syntactic structure to the resulting parses as can be seen in Table 2. With the additional noun phrase bracketing, the resulting Bleu score increased 0.23 points or 2.43%. The improvement is statistically significant with 95% confidence using pairwise bootstrapping of 1,000 test sets randomly sampled with replacement (Koehn, 2004; Zhang et al., 2004). In Figure 4 we can see that the difference between each of the 1,000 samples was above 0, meaning the Gold NP Parser performed consistently better given each sample.

Systems	Bleu
Baseline Parser	9.47
Gold NP Parser	**9.70**

Table 2: TectoMT results of a complete system run with both the Baseline Parser and Gold NP Parser. Both are tested on WMT08 data. Results are an average of 1,000 bootstrapped test sets with replacement.

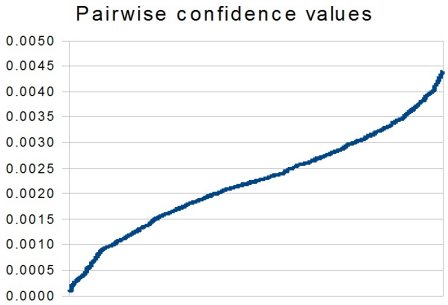

Figure 4: The Gold NP Parser shows statistically significant improvement with 95% confidence. The difference in Bleu score is represented on the Y-axis and the bootstrap iteration is displayed on the X-axis. The samples were sorted by the difference in bleu score.

Visually, changes can be seen in the English side parse that affect the overall translation quality. Sentences that contained incorrect noun phrase structure such as "The second vice-president and Economy minister, Pedro Solbes" as seen in Figure 5 and Figure 6 were more correctly parsed in the Gold NP Parser. In Figure 5 "and" is incorrectly assigned to the bottom of a noun phrase and does not connect any segments together in the output of the Baseline Parser, while it connects two phrases in Figure 6 which is the output of the Gold NP Parser. This shift in bracketing also allows the proper noun, which is shaded, to be assigned to the correct head, the rightmost noun in the phrase.

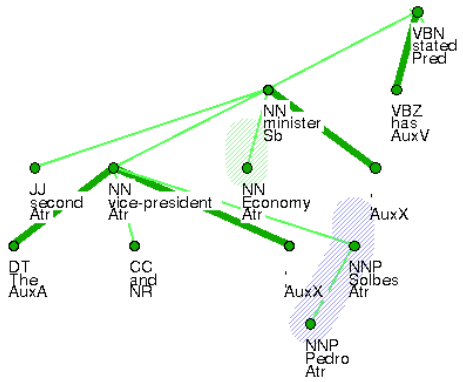

Figure 5: The parse created with the data with flat structures does not appear to handle noun phrases with more depth, in this case the 'and' does not properly connect the two components.

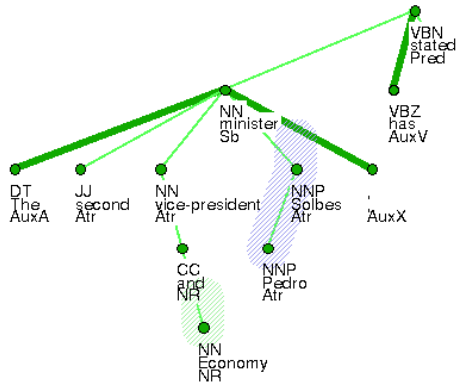

Figure 6: With the addition of noun phrase structure in parser, the complicated noun phrase appears to be better structured. The 'and' connects two components instead of improperly being a leaf node.

5 Conclusion

This paper has demonstrated the benefit of additional noun phrase bracketing in training data for use in dependency parsing and machine translation. Using the additional structure, the dependency parser's accuracy was minimally reduced. Despite this reduction, machine translation, much further down the NLP pipeline, obtained a 2.43% jump in Bleu score and is statistically significant with 95% confidence. Future work should examine similar experiments with MaltParser and other machine translation systems.

6 Acknowledgements

This research has received funding from the European Commissions 7th Framework Program (FP7) under grant agreement n° 238405 (CLARA), and from grant MSM 0021620838. I would like to thank Zdeněk Žabokrtský for his guidance in this research and also the anonymous reviewers for their comments.

References

Sabine Buchholz and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06, pages 149–164, Morristown, NJ, USA. Association for Computational Linguistics.

Michel Galley and Christopher D. Manning. 2009. Quadratic-time dependency parsing for machine translation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 773–781, Suntec, Singapore, August. Association for Computational Linguistics.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for English. In *Proceedings of NODALIDA 2007*, pages 105–112, Tartu, Estonia, May 25-26.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: the penn treebank. *Comput. Linguist.*, 19:313–330, June.

Ryan McDonald and Joakim Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 122–131.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 523–530, Morristown, NJ, USA. Association for Computational Linguistics.

Joakim Nivre. 2003. An efficient algorithm for projective dependency parsing. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT*, pages 149–160.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Martin Popel, Zdeněk Žabokrtský, and Jan Ptáček. 2010. Tectomt: Modular nlp framework. In *IceTAL*, pages 293–304.

Petr Sgall. 1967. *Generativní popis jazyka a česká deklinace*. Academia, Prague, Czech Republic.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The conll-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*, CoNLL '08, pages 159–177, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Vadas and James Curran. 2007a. Adding noun phrase structure to the penn treebank. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 240–247, Prague, Czech Republic, June. Association for Computational Linguistics.

David Vadas and James R. Curran. 2007b. Large-scale supervised models for noun phrase bracketing. In *Conference of the Pacific Association for Computational Linguistics (PACLING)*, pages 104–112, Melbourne, Australia, September.

David Vadas and James R. Curran. 2007c. Parsing internal noun phrase structure with collins' models. In *Proceedings of the Australasian Language Technology Workshop 2007*, pages 109–116, Melbourne, Australia, December.

David Vadas and James R. Curran. 2008. Parsing noun phrase structure with CCG. In *Proceedings of ACL-08: HLT*, pages 335–343, Columbus, Ohio, June. Association for Computational Linguistics.

Zdeněk Žabokrtský, Jan Ptáček, and Petr Pajas. 2008. Tectomt: highly modular mt system with tectogrammatics used as transfer layer. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 167–170, Morristown, NJ, USA. Association for Computational Linguistics.

Ying Zhang, Stephan Vogel, and Alex Waibel. 2004. Interpreting bleu/nist scores: How much improvement do we need to have a better system. In *In Proceedings of Proceedings of Language Resources and Evaluation (LREC-2004*, pages 2051–2054.