Running into Brick Walls Attempting to Improve a Simple Unsupervised Parser

Martin Riedl, Tim Feuerbach and **Chris Biemann**
Language Technology Group, CS Department, TU Darmstadt, Germany
riedl@cs.tu-darmstadt.de, uni@spell.work, biem@cs.tu-darmstadt.de

Abstract

In this article, we present a re-implementation of a simple unsupervised parser introduced by Søgaard (2012). This parser is able to parse sentences without any training. Furthermore, we propose various extensions to this parser. We evaluate the impact of several extensions on six languages. While we observe some improvements, different extensions impact different languages differently and we cannot give language-independent recommendations.

1 Introduction

Syntactic dependency parsing is a major preprocessing step needed for most applications and tasks in natural language processing like question answering (Hirschman and Gaizauskas, 2001), machine translation or similarity computations, e.g. (Levy and Goldberg, 2014; Weeds et al., 2004; Curran and Moens, 2002). However, most available dependency parsers are based on supervised machine learning algorithms, which need to be trained on manually created data. In addition, the creation of such training data is time-consuming and larger treebanks are not available for many languages.

In Riedl et al. (2014) several unsupervised dependency parsers have been extrinsically evaluated by using them as context representations for computing distributional similarities. In this work, the unsupervised parser by Søgaard (2012) yielded the second best results while being the fastest parser. In contrast to the other unsupervised dependency parsers, it does not require any training on raw text and is able to perform the parsing sentence-wise as opposed to whole-corpus parsing.

Whereas some unsupervised dependency parsers, e.g. Klein and Manning (2002), have been optimized and extended, e.g. Gillenwater et al. (2010), no further extensions have been proposed to many other unsupervised dependency parsers.

As the parser introduced by Søgaard (2012) is very basic in its heuristics, we will investigate whether integrating further features can improve its parsing performance. For this, we consider using semantics and Multiword Expressions (MWEs). Additionally, we re-run the parsing and train a supervised parser based on the output of the unsupervised parser.

2 Related Work

One of the first unsupervised syntactic dependency parsers that outperformed a random baseline was introduced by van Zaanen (2001) and uses an alignment-based learning approach. This algorithm is based on comparisons of sentences and uses sequence regularities in the corpus as constituents. A more sophisticated algorithm was presented by Klein and Manning (2002) that is based on an EM approach, which uses the linguistic phenomenon that long constituents often have shorter representations of the same grammatical function when they occur within a similar context. A combination of the work of Klein and Manning (2002) with a dependency model was presented by Klein and Manning (2004), which is called Dependency Model with Valence (DMV). This approach was the first one that outperformed the right branching baseline. Due to these results, this model has been extended by using lexical information (Headden III et al., 2009) and adding posterior regularizations in the training process (Gillenwater et al., 2010). These approaches require training, based on raw text or POS-tagged text. In contrast the method introduced by Søgaard (2012) does not require any training and can be applied with and without POS information.

Information about Multi-word Expressions (MWEs) has been shown to be beneficial for supervised dependency parsers. Le Roux et al. (2014) showed that for French, the detection of MWEs improves the parsing performance. Similarly, Eryiğit

et al. (2011) demonstrated that predicting Multiword Expressions (MWEs) and using such information for training a parser increases the performance.

3 Søgaard's Parser

In this paper we extend the unsupervised parser introduced in Søgaard (2012). It operates on single sentences and has three stages. First, tokens are ranked according to their valency. This is achieved by creating a multigraph with the sentence's tokens as its nodes. Edges are added following these heuristics:

- add pairwise edges to any neighbor in 1-step vicinity
- add pairwise edges to any neighbor in 2-step vicinity
- add an edge to a function word (determined by a word list) from any 1-step neighbor. The function word list is generated in advance using a simplification of TextRank (Mihalcea and Tarau, 2004) without stopword removal. The method is applied to the training data and we extract the top 50 words.
- add an edge to the verb from every other token in the sentence
- add pairwise edges between any tokens for which the 3-letter-prefix does not match
- add pairwise edges between any tokens for which the 3-letter-suffix does not match

Then, PageRank (Brin and Page, 1998) is applied in order to rank the nodes. The tokens are sorted in descending order to their rank and stored in a list called *dependents*. Additionally, a list called *head nodes* is created and a *ROOT* node is added. At the final stage, the dependency tree is created according to the following algorithm:

- while *dependents* is not empty
  1. remove first token
  2. assign a head from *head nodes*:
     - if universal dependency rules (Naseem et al., 2010) are used: assign the closest head (in terms of distance in the sentence) for which a rule fires
     - else, or if no rule applies: assign the closest head candidate
     - if ties: assign the head with the highest PageRank score
  3. add token to *head nodes*

4 Extensions

In this section, we describe all the extensions we will apply in order to achieve improvements for the parsing.

4.1 Re-running the Parsing

We expect that dependencies produced by the unsupervised parser might be helpful also for the parsing. Thus, we first apply Søgaard's parser to a new sentence. Then, we add the detected syntactic dependencies as weights to the normal heuristics, apply the ranking and build the dependency tree again.

4.2 Learning Regularities

One main advantage of Søgaard's parser is that it does not require any training since it is applies a collection of heuristics. However, previous decisions provide valuable information about the relationship of various POS. In order to utilize this information, we apply Søgaard's parser on raw text and use the dependency labels as training data for the MaltParser (Nivre, 2008). Using this model, we parse the test data and perform the evaluation on these dependencies.

4.3 Integrating Semantics

Words that have a similar meaning are usually on a similar level of salience. Therefore, we experimented removing edges between neighboring tokens that have a distributionally similar meaning. We use similarities computed with the approach by Biemann and Riedl (2013). As context representation we use the so-called trigram context extraction method, which uses the left and right neighboring word as context. In addition, we show results for German and English when using similarities computed using syntactic dependencies from a supervised method as context.

4.4 Integrating Multiword Expressions

Recognizing MWEs is beneficial for parsing, cf. Le Roux et al. (2014). Thus, we add edges between words that are recognized as MWEs according to a generated list of MWEs. This resource is generated using the unsupervised word sequence ranking measure called DRUID (Riedl and Biemann, 2015).

The measure does not require any POS filtering and can be applied to corpora without any linguistic pre-processing. We computed DRUID on a larger background corpus and used only word sequences of a maximum length of 4 and a score above 0.5. If a token was part of the same MWE as a head candidate, we preferred that candidate in the same vein as if it would match a universal rule.

5 Experimental Setting

We evaluate on German, Danish, Dutch, Portuguese, and Swedish test data from the 2006 CoNLL shared task on multi-lingual dependency parsing[1]. For English, we evaluated on Section 23 of the Wall Street Journal part of Penn Treebank III (PTB-III). As development set we use Section 11 of PTB-III. The treebank was converted to dependencies using the LTH Constituent-to-Dependency converter[2]. We train the MaltParser based on the parser's output on the train data of Danish, Dutch, German, Portuguese and Swedish. For English, we used the entire Wall Street Journal section of PTB-III. Unlabeled attachment scores were obtained using the official CoNLL-07 scorer.

For computing the similarities and the MWE resource for English we use 105M sentences of newspaper extracted from the Leipzig Corpora Collection (LCC) (Richter et al., 2006) and Gigaword (Parker et al., 2011). The computations for German are performed on 70M sentences from the LCC; for Swedish 60M sentences of newspaper data from Spraakbanken[3] are used. For Dutch, we compute similarities and MWEs based on 259 million sentences from the Dutch web corpus (Schäfer and Bildhauer, 2013).[4] The Portuguese is computed based on the Brazilian web corpus (Boos et al., 2014).

The dependency-based similarities are computed using the Stanford Parser (de Marneffe et al., 2006) for English and the MaltParser (Nivre, 2008) for German.

6 Results

In this section, we show the result of our re-implementation and additionally show the performances on different languages when incorporating the different modifications.

6.1 Performance on several languages

The results with our implementation[5] are presented for the six languages in Table 1, next to the results from Søgaard (2012).

	no UR		UR			
	We	Søgaard	We	Søgaard	Baseline	Oracle
Danish	55.70	50.8	54.38	51.4	43.77	71.49
Dutch	40.85	39.7	40.45	38.3	36.21	65.38
English	43.29	52.6	52.00	59.9	26.38	76.13
German	44.73	48.7	55.15	57.6	25.61	69.85
Portuguese	39.07	47.0	48.75	54.6	34.22	70.45
Swedish	47.68	52.3	56.86	60.5	30.60	71.87

Table 1: Basic unlabeled attachment scores on sentences with at most 10 tokens without punctuation. UR: Universal dependency rules enabled.

For unknown reasons, we cannot replicate results reported in (Søgaard, 2012)[6]. Whereas for Danish and Dutch, we observe higher scores than the ones in the paper, most results are below the performance of Søgaard (2012). This finding is consistent for both using universal dependency rules (URs) and without using URs. In accordance with the original implementation, our re-implementation outperforms the right-branching baseline. Like Søgaard (2012), we considered as upper bound an oracle function that ranks tokens in a top-to-bottom, left-to-right fashion according to their gold dependency trees.

6.2 Performance of Extensions

In this section, we describe the performance of the various extensions for adding edges into the graph-based method. First, we show results in Table 2 when re-running the algorithm, using dependency links from the first pass as additional edges. The number of additional edges (6) was determined using the English development data.

We observe that this extension reduces the performance both for Danish and Dutch tremendously. However, for English we observe significant improvements both for using/not using universal dependency rules. For German and Portuguese we only observe improvements when using universal

[1] http://ilk.uvt.nl/conll/post_task_data.html

[2] http://nlp.cs.lth.se/software/treebank-converter

[3] http://spraakbanken.gu.se

[4] available at: http://webcorpora.org/.

[5] The implementation is available under the Apache 2.0 license: http://jobimtext.org/jobimtext//components/unsupervised-parser

[6] Although we also tested the original implementation, we could not achieve the results from the paper. This might be attributed due to different keyword lists and different corpus transformations.

	no UR		UR	
	Basic	Re-running	Basic	Re-running
Danish	55.70	53.58	54.38	50.66
Dutch	40.85	36.21	40.45	35.81
English	43.29	43.62†	52.0	53.15†
German	44.73	44.36	55.15	58.33†
Portuguese	39.07	38.90	48.75	50.25
Swedish	47.68	47.29	56.86	56.17

Table 2: Results for re-running the algorithm on the same sentence. Scores with a † are significant over the basic score (paired bootstrap resampling test (Koehn, 2004) with $p = 0.05$, $n = 1000$).

dependency rules. Thus, no general trend can be obtained for re-using unsupervised dependency information.

Next, we show results in Table 3 when using the links obtained with Søgaard's dependency parser in order to train the supervised MaltParser as described in Section 4.2. Except for Danish, this

	no UR		UR	
	Basic	+MaltParser	Basic	+MaltParser
Danish	55.70	54.91	54.38	54.51
Dutch	40.85	41.25	40.45	43.77†
English	43.29	44.51†	52.0	50.19
German	44.73	45.47	55.15	54.53
Portuguese	39.07	39.40	48.75	46.08
Swedish	47.68	48.86	56.86	55.48

Table 3: Results for using the unsupervised dependency parses for training MaltParser and using MaltParser to parse the test data.

approach consistently yields improvements. This changes when universal rules are used; here, the performance on Dutch and Danish increases. For English we significantly outperform the basic results. However this comes at the cost of losing the runtime benefit of Søgaard's parser.

Next, we present the impact when integrating semantic information and MWE information into the unsupervised parser. As can be obtained from Table 4, using semantic information that is computed

	no UR			UR		
	Basic	MWEs	Semantics	Basic	MWE	Semantics
Dutch	40.85	40.98	40.72	40.45	40.58	40.05
English	43.29	43.33	43.03	52.0	51.96	52.15
German	44.73	44.98	44.61	55.15	54.90	55.64
Portuguese	39.07	39.23	39.40	48.75	48.41	49.42†
Swedish	47.68	47.78	47.09	56.86	56.47	56.37

Table 4: Results for using semantic information and preferring heads from the same MWE.

on neighboring words decreases the performance for all languages but Portuguese. Applying these rules, we observe declines for Dutch and Swedish, but gain improvements for the remaining languages. Additionally, we tested similarities for English and German that are computed using syntactic dependencies as context representation for testing purposes, as it defies the goal of inducing a parser for languages without treebank resources. Without using universal rules, we observe a decrease in terms of performance for English (43.25) and obtain slight increases for German (45.22).

Integrating information from the MWE resource and not applying the universal rules results in consistent yet small improvements among all tested languages (see Table 4). Similar to the results using semantic information, scores increase for all languages except for Dutch when using universal rules.

In the next experiment, we combined several extensions. As can be observed from Table 5 integrating semantic and MWE information improves the performance in all cases except for Swedish. In addition we also present results when adding the re-running to the algorithm. For Dutch and Swedish we notice a performance decline. When using universal rules, we observe an increase in performance for English, German, and Portuguese.

	no UR			UR		
	Basic	MWEs +Sem	MWEs +Sem +Re-running	Basic	MWEs +Sem	MWEs +Sem +Re-running
Dutch	**40.85**	**40.85**	35.94	40.45	40.45	35.15
English	43.29	43.37	43.37	52.0	**52.11**	**52.11**
German	44.73	45.34	44.73	55.15	55.51	**58.46**†
Portuguese	39.07	39.57†	39.57	48.75	49.08	**50.92**
Swedish	47.68	47.19	46.40	**56.86**	55.97	55.08

Table 5: Results for combining some of the extensions.

7 Conclusion

In this paper we have shown that intuitive and reasonable extensions for Søgaard's dependency parser do not translate into general improvements among all languages. This is in line with the findings described in (Riedl et al., 2014) that most unsupervised dependency parsers are optimized for English rather than the other languages. Whereas some extensions yield minor improvements, we cannot give any language-independent recommendation.

References

Chris Biemann and Martin Riedl. 2013. Text: Now in 2D! A Framework for Lexical Expansion with Contextual Similarity. *Journal of Language Modelling*, 1(1):55–95.

Rodrigo Boos, Kassius Prestes, Aline Villavicencio, and Muntsa Padró. 2014. In Jorge Baptista, Nuno Mamede, Sara Candeias, Ivandré Paraboni, Thiago A. S. Pardo, and Maria das Graças Volpe Nunes, editors, *Proceedings of the 11th International Conference on Computational Processing of the Portuguese Language*, PROPOR 2014, pages 201–206, São Carlos/SP, Brazil.

Sergey Brin and Lawrence Page. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Seventh International World-Wide Web Conference*, WWW 1998, pages 107–117, Brisbane, Australia.

James R. Curran and Marc Moens. 2002. Improvements in automatic thesaurus extraction. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition - Volume 9*, ULA 2002, pages 59–66, Philadelphia, PA, USA.

Marie-Catherine de Marneffe, Bill Maccartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the International Conference on Language Resources and Evaluation*, LREC 2006, pages 449–454, Genova, Italy.

Gülşen Eryiğit, Tugay İlbay, and Ozan Arkan Can. 2011. Multiword expressions in statistical dependency parsing. In *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*, SPMRL 2011, pages 45–55, Dublin, Ireland.

Jennifer Gillenwater, Kuzman Ganchev, João Graça, Fernando Pereira, and Ben Taskar. 2010. Sparsity in dependency grammar induction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics - Short Papers*, ACL 2010, pages 194–199, Uppsala, Sweden.

William P. Headden III, Mark Johnson, and David McClosky. 2009. Improving unsupervised dependency parsing with richer contexts and smoothing. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL-HLT 2009, pages 101–109, Boulder, CO, USA.

Lynette Hirschman and Rob Gaizauskas. 2001. Natural Language Question Answering: The View from Here. *Journal of Natural Language Engineering (NLE)*, 7(4):275–300.

Dan Klein and Christopher D. Manning. 2002. A generative constituent-context model for improved grammar induction. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL 2002, pages 128–135, Philadelphia, PA, USA.

Dan Klein and Christopher D. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL 2004, pages 478–485, Barcelona, Spain.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2004, pages 388–395, Barcelona, Spain.

Joseph Le Roux, Antoine Rozenknop, and Matthieu Constant. 2014. Syntactic parsing and compound recognition via dual decomposition: Application to french. In *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers*, COLING 2014, pages 1875–1885, Dublin, Ireland.

Omer Levy and Yoav Goldberg. 2014. Dependency-Based Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, ACL 2014, pages 302–308, Baltimore, MD, USA.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2004, pages 404–411, Barcelona, Spain.

Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. 2010. Using universal linguistic knowledge to guide grammar induction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP 2010, pages 1234–1244, Cambridge, MA, USA.

Joakim Nivre. 2008. Algorithms for deterministic incremental dependency parsing. *Computational Linguistic*, 34(4):513–553.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. *English Gigaword Fifth Edition*. Linguistic Data Consortium, Philadelphia, PA, USA.

Matthias Richter, Uwe Quasthoff, Erla Hallsteinsdóttir, and Chris Biemann. 2006. Exploiting the Leipzig Corpora Collection. In *Proceedings of the Fifth Slovenian and First International Language Technologies Conference*, IS-LTC 2006, pages 68–73, Ljubljana, Slovenia.

Martin Riedl and Chris Biemann. 2015. A Single Word is not Enough: Ranking Multiword Expressions Using Distributional Semantics. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP 2015, pages 2430–2440, Lisboa, Portugal.

Martin Riedl, Irina Alles, and Chris Biemann. 2014. Combining supervised and unsupervised parsing for distributional similarity. In *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers*, COLING 2014, pages 1435–1446, Dublin, Ireland.

Roland Schäfer and Felix Bildhauer. 2013. *Web Corpus Construction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

Anders Søgaard. 2012. Unsupervised dependency parsing without training. *Natural Language Engineering*, 18(02):187–203.

Menno van Zaanen. 2001. Building treebanks using a grammar induction system. Technical report, University of Leeds, UK, School of Computer Studies.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING 2004, pages 1015–1021, Geneva, Switzerland.