ITAT

Twelve Years of Unsupervised Dependency Parsing

David Mareček

Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics
Charles University in Prague, Malostranské nám. 25, 118 00, Prague, Czech Republic
`marecek@ufal.mff.cuni.cz`

Abstract: In the last 12 years, there has been a big progress in the field of unsupervised dependency parsing. Different approaches however sometimes differ in motivation and definition of the problem. Some of them allow using resources that are forbidden by others, since they are treated as a kind of supervision. The goal of this paper is to define all the variants of unsupervised dependency parsing problem and show their motivation, progress, and the best results. We also discuss the usefulness of the unsupervised parsing generally, both for the formal linguistics and for the applications.

1 Introduction

Dependency parsing is one of the traditional tasks in natural language processing. It gets a tokenized sentence as input (in most cases, individual tokens (words) are labelled by part-of-speech (POS) tags), and produces a rooted dependency tree, in which the nodes correspond to words and edges correspond to syntactic relations between the words.

Rule-based approaches of dependency parsing were suppressed by the statistical dependency parsers, which achieved better quality compared to the human annotations. Important milestones in dependency parsing were the CoNLL shared tasks in 2006 [4] and 2007 [22]. They provided about 20 treebanks of different languages available in the same format. This became the standard for measuring quality of dependency parsers in fact up to now. [1]

At the same time, there were efforts to develop a parser that does not need any annotated data. The unsupervised parsers infer the dependency structures based on language- and tagset-independent properties of dependency trees, which is mainly the low entropies of the governing-dependent word pairs and low entropies of the word fertilities (number of dependents).

One general motivation is to be able to parse languages for which no annotated treebanks exist. Less sound motivation is to create a dependency structure which better suits a particular NLP application, e.g. machine translation.

This is a survey paper about unsupervised dependency parsers. Since different approaches have different moti-

vations, allow to use different kinds of data and different amount of knowledge about them, they cannot be compared because of different degree of (un)supervision. The aim of this paper is to cluster the approaches to several groups in which they are comparable and to show the most important ones together with the results.

The paper is structured as follows: In Section 2, we define different unsupervised parsing problem settings and summarize the motivations and advantages. Section 3 describes different evaluation measures developed for unsupervised parsers. In Section 4, we go through the works done in this field and describe the most important approaches. In Section 5, we compare the results across data, parsers, and languages. Section 6 discusses generally the usefulness of unsupervised parsing methods in linguistics and in applications. Section 7 concludes.

2 Problem Settings

Some unsupervised parsing approaches use different kinds of data for the grammar inference than others. What is used in one is treated as not allowed kind of supervision in another. We therefore categorize the approaches into four groups. They are described in the following subsections and sorted from the least unsupervised (more data and knowledge) to the most unsupervised (less data and knowledge).

2.1 Using supervised POS tags and some knowledge about them

In the first group of approaches, there are parsers that need the sentences labelled by supervised POS tags, i.e. by a manually designed tagset. On top of that they also somehow utilizes the knowledge about the tagset. For example, they know which tags are used for verbs and therefore treat them differently through the grammar inference. This is the main difference from the second group (Section 2.2) and it is sometimes considered as a bit of cheating. If we know the meaning of the POS tags, we could easily build a simple rule-based parser, which would definitely not be unsupervised. This also relates to so-called delexicalized parsing, where the parser is trained on a different language with the same POS tagset and the model can then be used for languages without treebanks. This is however beyond the scope of this paper. The approaches we assigned to this group however use only a bit of such knowledge that help the inferred structures to be in a required shape.

---

[1] In recent years, many researchers work on a project called Universal Dependencies [21], a collection of treebanks for many languages (51 treebanks and 40 languages in its current version 1.3), where the morphological and dependency annotation styles are unified across the languages.

2.2 Using supervised POS tags without any knowledge about them

The majority of works describing unsupervised dependency parsing utilize supervised POS tags without any knowledge about them. In other words, the parsers take the POS tags only as labels without any meaning. It is a bit strange not to tell the parser anything, for example: "ADJ are adjectives and often depends on the following NOUNs", if there is such possibility, however allowing it would bring the parsers to the first group (Section 2.1) whose unsupervisedness is sometimes disputable.

Nevertheless, what is more strange, is the usage of supervised POS tags. The POS tags carry a lot of syntactic information. Imagine a sequence of POS tags "ADJ NOUN VERB PREP ADJ NOUN". You would easily build the most probable dependency tree. The motivation of this problem setting may be:

1. We want to compare supervised and unsupervised parsers operating on the same tagset.

2. We want to evaluate an unsupervised parser and, in the future, we will use the unsupervised word classes instead of the supervised tags on low-resourced languages and hope that it will work as well.

3. We have a language without treebank and we have a POS tagger. However, we are not able to find the meaning of POS tags used.

The third option is rather hypothetical. We always find someone who speaks that language or have a parallel corpus from which we could get basic meanings of individual words and tags.

It is also worth to mention that almost all the experiments and evaluation in the papers were done using gold standard POS tags, i.e. the POS tags assigned manually by human annotations. This is not surprising. While the qualities of unsupervised parsers are substantially lower than the qualities of supervised parsers, it is not worthy to make experiments also with the predicted POS tags.

2.3 Using unsupervised POS tags

The lower attention was given to fully unsupervised parsers using unsupervised POS tags (words classes). The only source they use is raw text. The motivation is obvious here: If we want to analyze a language without any manually annotated resources, we need exactly this approach. Other motivation could be the need of having different structures from that present in annotated treebanks. Majority of works here used the same parser as for supervised POS tags (Section 2.2) and obtain the unsupervised POS tags by some of the best word clustering tools available.

2.4 Direct parsing from raw text without POS tags

The last setting we describe is unsupervised parsing from raw texts. Here we do not use any POS tags or word classes. The only units the parser plays with are the words. The results should be theoretically compared with the previous category (2.3), where unsupervised word clustering is used. However, the word classes are typically inferred on much larger text corpora than dependency trees are. This approaches use therefore much less data for the inference and that is why we assign them into a separate category. Such approaches would be the most elegant way of parsing, however, they naturally achieve very poor results.

3 Parsing Evaluation

The unsupervised parsing approaches sometimes differ also in evaluation metrics. The standard attachment score is sometimes found too strict to evaluate the inferred structures and therefore new, more tolerant metrics, are designed. The following three evaluation metrics exists:

1. **Directed attachment score** (unlabeled attachment score[2]) is a standard metric for measuring dependency parsing quality. It is a percentage of words correctly attached to their parents. It does not allow even the slightest local structural differences, which might be caused just by more or less arbitrary linguistic or technical conventions.

2. **Undirected attachment score** disregards the directions of edges and is therefore less biased towards such conventions. For example, there is no difference whether the parser attaches prepositions to nouns or nouns to prepositions. Nevertheless, this holds for all edges, including these with undoubted directions.

3. **Neutral edge direction**[3] metric proposed by [24] is even more tolerant in assessing parsing errors than the undirected attachment score. It treats not only node's parent and child as the correct answer, but also its grandparent.

Even though the alternative scores were proposed and sometimes used, the majority of experiments were evaluated by the directed attachment scores, probably because of its simplicity and the tradition in the field and also because the other two did not prove to be substantially better.

4 Unsupervised Dependency Parsers

In this Section, we summarize and describe the most important works in the field of unsupervised dependency

---

[2]We do not want to use the abbreviation UAS for the *unlabeled attachment score* here, since it could be mistaken for *undirected attachment score*.

[3]http://www.cs.huji.ac.il/ roys02/softwae/ned.html

parsing through the last 12 years. Even though there were a couple of works before, the first paper with results better than a chain baseline [4] was the Dependency Model with Valence by Klein and Manning [13].

We first describe the methods using supervised POS tags without any other knowledge (Section 2.2) in Sections 4.1 and 4.2, then we switch to other settings. A detailed table with results over different methods and different problem settings is shown in Section 5.

4.1 Dependency Model with Valence

We start with Dependency Model with Valence (DMV), which was introduced by Klein and Manning [13]. It is the most popular approach, which was followed by many other researchers and improved in many ways. It is a generative model that generates dependency trees using two submodels:

- Stop model $p_{stop}(\cdot|t_g, dir)$ represents probability of not generating another dependent in direction dir to a node with POS tag t_g. The direction dir can be left or right. If $p_{stop} = 1$, the node with the tag t_g cannot have any dependent in direction dir. If it is 1 in both directions, the node is a leaf.

- Attach model $p_{attach}(t_d|t_g, dir)$ represents probability that the dependent of the node with POS tag t_g in direction dir is labeled with POS tag t_d.

The grammar consisting of probability distributions p_{stop} and p_{attach} is learned using the Expectation Maximization inside-outside algorithm [12]. The learning is further improved by Smith et al. [26] and Cohen et al. [8]. Headden et al. [11] introduce the Extended Valence Grammar and add lexicalization and smoothing. Besides the POS tags, the parser begin to operate with word forms as well. Blunsom and Cohn [2] use tree substitution grammars, which allow learning of larger dependency fragments by employing the Pitman-Yor process. Spitkovsky [30] improves the inference using iterated learning of increasingly longer sentences. Further improvements are achieved by better dealing with punctuation [32] and new "boundary" models [33]. Spitkovsky also improves the learning itself in [31] and [34].

Mareček and Straka [16] use so called reducibility principle to predict p_{stop} probabilities for individual POS tags from raw texts, add it to the Dependency Model with Valence and use Gibbs sampling to infer the grammar. In [19], they suppose that the function words, which can be predicted by they shortness, have fixed low number of dependents and move the parsing results even a bit higher.

4.2 Other approaches using supervised POS tagset

There are also approaches not based on DMV, even though their models are not far from it. Mareček and Žabokrtský [18] use a fertility to model number of children for particular POS tags instead of the p_{stop} model.

Sogaard [27] explores a completely different view in which a dependency structure is among other things a partial order on the nodes in terms of centrality or saliency.

Cohen et al. [7] do the grammar inference multilingually on more languages. The data do not need to be parallel, they only have to share the tagset. The inference is then less prone to skew to bad solutions due to the language differences.

Bisk and Hockenmaier [1] use the Combinatory Categorial Grammars for dependency structure induction.

4.3 Approaches using some knowledge about the POS tags

The "less unsupervised" approaches utilizing an external knowledge of the POS tagset reach often better attachment scores than the previous approaches. Any additional knowledge about the tags used can be very strong and can change the inferred structures dramatically. For example, Naseem et al. [20] follow Eisner [9] and make use of manually-specified universal dependency rules such as *Verb→Noun* or *Noun→Adjective* to guide grammar induction and improve the results by a wide margin. Mareček and Žabokrtský [17] show that only the information that "the POS tags for nouns are more frequent than the POS tags for verbs" very much improves the baseline. This however fails for example in case the POS tags for nouns are subcategorized in some way. Then we would need to know which POS tags are for nouns and group them together. Rasooli and Faili [23] identify the last verb in the sentence, minimize its probability of reduction and push it to the root position, and also make a huge improvement.

Such approaches achieve better results; however, they are useless for grammar induction for languages, for which the tagger is not available.

4.4 Approaches using unsupervised POS tags

These approaches mostly do not bring any new methods. The authors only take their unsupervised parsers we presented in Section 4.1, take a word clustering tool to produce unsupervised POS tags and run their parser on them. Spitkovsky [29] took the clustering tool by Clark [6] and Brown et al. [3] and showed that the parsing with supervised POS tags can be outperformed for English, if the word classes are used instead. Mareček [15] performed similar experiments on 30 languages and showed that on some of them the use of unsupervised word classes instead of supervised POS tags improve the parsing accuracy. The average score across the languages was however significantly worse.

---

[4] In the left or right chain baseline, each word is attached to the next or previous one respectively.

Christodoulopoulos et al. [5] try to do inference of POS tags and dependency structure together. After random initialization, they alternate the prediction of the structure based on the POS tags and prediction of the POS tags based on the structure.

4.5 Approaches using raw text only

There are couple of approaches, which do not need any word categorization. We only mention the incremental parsing by Yoav Seginer [25]. His algorithm collects lists of labels for each word, based on neighboring words, and then directly uses these labels to parse.

5 Results

In Tables 1, 2, and 3, we summarize the results over the individual parsers, data, and settings. Unfortunately, different parsers were evaluated on different data. In the beginnings, the parsers were evaluated mainly on the English Penn Treebank [14] (transformed to dependencies) and some only on the short sentences of length up to 10 (ptb10), since the shorter sentences were easier to parse and the resulting scores did not look so bad. See Table 1.

After the unsupervised parsers were improved and achieved much better results than simple baselines, they started to be evaluated across languages and on sentences of all lengths (Table 2).

In 2012, there has been a shared task on unsupervised dependency parsing named "The PASCAL challenge on Grammar Induction" [10]. Seven competing parsers were evaluated on new datasets comprising ten different languages, including simpler English used by small children. See Table 3.

Unfortunately, some of the parsers were evaluated on non-standard data or with non-standard metrics and therefor their results could not be added into any of the three tables.

All the tables share the same format: each method is labelled by a link to the references and by a group label: *SP* for using supervised POS tags, *UP* for using unsupervised POS tags, and *SP+K* when an additional knowledge about the supervised tags was used.

6 Usefulness of Unsupervised parsers in linguistics and applications

We could see a lot of work done in the field of unsupervised parsing in the last 12 years. The quality of induced structures are better than before, but the supervised parsers are still better then the unsupervised ones by a wide margin. However, for low resourced languages, for which no annotated data exists, this is the way, how to obtain their syntactic structure.

A more serious problem with unsupervised parsing is that, according to our knowledge, there were so far no

works incorporating any kind of unsupervised parsing into applications, even though many papers mention that in some cases, an unsupervised structures, different from manual annotations following a given schema, may be very beneficial.

Moreover, in the last two years, no new strong paper about unsupervised parsing appeared in NLP conferences. Instead, a new techniques have arrived: The recurrent neural networks, which may fulfill the previous motivations for unsupervised parsing – to find a structure of language that would help machines to understand it better. Instead of dependency trees, some structures are hidden in hidden states of the deep neural networks.

From the linguistic point of view, the structures inferred by unsupervised parsers can be compared to the manually annotated treebanks. What are the differences? How the unsupervised methods deal with phenomena that are not clear how to parse? Should prepositions depend on nouns or vice versa? And what about coordinations? Many such questions could be answered, however, neither this topic was studied so far.

7 Conclusions

We categorized the unsupervised dependency parsers into four groups according to their needs of data, so that they could be fairly compared. We make a survey over the most important papers and works that reached state-of-the-art results when they were published. We showed a comparison of the results across the methods and languages. It is apparent that there is a big variance over the attachment scores for individual languages. The good performance of a method on one language tells nothing about the performance on another language. We hope that this paper brings to readers some system in the world of unsupervised parsing.

Acknowledgments

References

[1] Yonatan Bisk and Julia Hockenmaier. Induction of linguistic structure with combinatory categorial grammars. *The NAACL-HLT Workshop on the Induction of Linguistic Structure*, page 90, 2012.

[2] Phil Blunsom and Trevor Cohn. Unsupervised induction of tree substitution grammars for dependency parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1204–1213, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

	left baseline	right baseline	Klein 2004	Cohen 2008	Headden 2009	Blunsom 2010	Spitkovsky 2010	Spitkovsky 2011	Spitkovsky 2011b
link	–	–	[13]	[8]	[11]	[2]	[30]	[32]	[29]
group	–	–	SP	SP	SP	SP	SP	SP	UP
PennTreebank (up to 10 words)	27.7	38.4	47.5	59.3	68.8	67.7	57.1	69.5	
PennTreebank (all sentences)	24.0	31.7		39.0		55.7	45.0	58.4	59.1

Table 1: Results comparison on Penn treebank [14] (automatically converted to dependencies by PennConverter).

	left baseline	right baseline	Spitkovsky 2011	Spitkovsky 2012	Spitkovsky 2013	Marecek 2012	Marecek 2013	Marecek 2014	Cohen 2011	Rasooli 2012	Marecek 2015	Marecek 2011
link	–	–	[32]	[33]	[34]	[18]	[16]	[19]	[7]	[23]	[15]	[17]
group	–	–	SP	SP	SP	SP	SP	SP	SP	SP+K	UP	SP+K
Arabic 2006	55.6	7.9	32.6	10.9	9.3	26.5	38.2	35.2				
Arabic 2007	59.0	6.0	35.3	44.9	26.8	27.9	35.3			52.1		24.8
Basque 2007	23.0	30.5	29.9	33.3	24.4	26.8	35.5	35.3			28.7	34.7
Bulgarian 2006	38.8	17.9	40.5	65.2	63.4	46.0	54.9	56.8	38.6	53.9	47.9	51.4
Catalan 2007	30.0	24.8	55.2	62.1	68.0	47.0	67.0	43.3			49.2	56.3
Chinese 2006	14.0	40.7	63.2	58.4								
Chinese 2007	13.4	41.3	57.0	52.5								
Czech 2006	26.9	25.2	37.8	55.1	44.0	49.5	52.4	55.6				
Czech 2007	29.6	24.2	36.2	54.2	34.3	48.0	51.9	54.3		42.4	38.0	33.3
Danish 2006	47.8	13.1	37.1	22.2	21.4	38.6	41.6	45.5	39.9	53.1	40.8	38.6
Dutch 2006	24.5	28.0	14.0	46.6	48.0	44.2	47.5	54.8	50.2	48.8	34.4	43.4
English 2007	21.0	29.4	50.3	29.6	58.2	49.2	55.4	58.5			47.9	23.8
German 2006	22.0	23.4	28.6	39.1	56.2	44.8	52.4	45.0			41.2	21.8
Greek 2007	19.7	31.4	21.2	26.9	45.4	20.2	26.3	34.5	38.9		39.7	33.4
Hungarian 2007	5.5	41.4	8.0	58.2	58.3	51.8	34.0	34.1			24.1	48.1
Italian 2007	37.4	21.6	28.8	40.7	34.9	52.3	39.4	51.0			41.4	60.6
Japanese 2006	13.8	67.2	27.5	22.7	63.0	50.8	61.2	61.9	61.7			53.5
Portuguese 2006	31.2	25.8	33.5	72.4	74.5	50.6	69.6	75.1	51.5	55.8	31.4	41.8
Slovenian 2006	26.6	24.3	31.2	35.2	50.9	18.1	35.7	47.3	36.0	22.4	18.3	34.6
Swedish 2006	27.8	25.9	46.4	50.7	49.7	48.2	54.5	55.6	39.9	34.3	54.9	26.9
Spanish 2006	29.8	24.7	32.3	28.2	61.4	51.9	61.1	61.7	40.5	31.2	60.1	54.6
Turkish 2007	1.6	65.5	40.9	34.4	37.9	15.7	56.9	57.0	38.6	27.4	55.0	32.1
average	26.4	29.8	33.2	42.9	48.6	40.0	48.7	50.0	43.3	43.0	38.9	39.4

Table 2: Results comparison on CoNLL datasets [4, 22]. Averaged numbers are sometimes not comparable since the average can be done over different set of languages.

	left baseline	right baseline	Blunsom 2010	Bisk 2012	Marecek 2012	Sogaard 2012	Sogaard 2012	Tu 2012	Klein 2004
link	–	–	[2]	[1]	[18]	[28]	[28]	[35]	[13]
group	–	–	SP	SP	SP	SP	SP+K	SP+K	SP
Arabic	9.3	64.8	61.1	47.2	64.8	47.2	54.6	66.7	45.4
Basque	34.3	24.4	56.3	50.3	30.3	33.3	22.3	58.6	43.2
Czech	28.9	34.3	50.0	48.5	57.5	45.5	51.2	59.0	31.2
Danish	18.7	49.2	46.2	49.3	51.3	56.9	60.5	60.8	50.2
Dutch	34.0	39.5	50.5	50.8	49.5	38.9	50.0	51.7	39.7
English (childes)	36.0	23.3	48.1	62.2	47.2	50.5	53.5	56.0	51.9
English (PTB)	40.4	19.9	72.1	73.7	67.4	44.8	61.0	74.7	44.7
Portuguese	28.1	37.7	54.3	76.3	59.9	47.7	71.1	55.7	37.2
Slovenian	35.9	14.7	65.8	53.9	51.4	39.7	50.3	67.7	37.2
Swedish	33.2	31.3	65.8	66.7	63.7	48.2	72.0	76.5	44.2
average	29.9	33.9	57.0	57.9	54.3	45.3	54.7	62.7	42.5

Table 3: Results comparison in the PASCAL Challenge on Grammar Induction [10].

[3] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, December 1992.

[4] Sabine Buchholz and Erwin Marsi. 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, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[5] Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. Turning the pipeline into a loop: Iterated unsupervised dependency parsing and PoS induction. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 96–99, June 2012.

[6] A. Clark. Combining distributional and morphological information for part of speech induction. *Proceedings of 10th EACL*, pages 59–66, 2003.

[7] Shay B. Cohen, Dipanjan Das, and Noah A. Smith. Unsupervised structure prediction with non-parallel multilingual guidance. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 50–61, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[8] Shay B. Cohen, Kevin Gimpel, and Noah A. Smith. Logistic normal priors for unsupervised probabilistic grammar induction. In *Neural Information Processing Systems*, pages 321–328, 2008.

[9] Jason Eisner. Three New Probabilistic Models for Dependency Parsing: An Exploration. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING-96)*, pages 340–345, Copenhagen, August 1996.

[10] Douwe Gelling, Trevor Cohn, Phil Blunsom, and Joao Graca. The PASCAL Challenge on Grammar Induction. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 64–80, Montréal, Canada, June 2012. Association for Computational Linguistics.

[11] William P. Headden III, Mark Johnson, and David McClosky. 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 '09, pages 101–109, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[12] Dan Klein. *The Unsupervised Learning of Natural Language Structure*. PhD thesis, Stanford University, 2005.

[13] Dan Klein and Christopher D. Manning. Corpus-based induction of syntactic structure: models of dependency and constituency. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[14] Mitchell P. Marcus, Beatrice Santorini, and Mary A. Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1994.

[15] David Mareček. Multilingual unsupervised dependency parsing with unsupervised pos tags. In Grigori Sidorov and N. Sofía Galicia-Haro, editors, *Advances in Artificial Intelligence and Soft Computing: 14th Mexican International Conference on Artificial Intelligence, MICAI 2015, Cuernavaca, Morelos, Mexico, October 25-31, 2015, Proceedings, Part I*, pages 72–82, Cham, 2015. Springer International Publishing.

[16] David Mareček and Milan Straka. Stop-probability estimates computed on a large corpus improve Unsupervised Dependency Parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 281–290, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[17] David Mareček and Zdeněk Žabokrtský. Gibbs Sampling with Treeness constraint in Unsupervised Dependency Parsing. In *Proceedings of RANLP Workshop on Ro-

bust Unsupervised and Semisupervised Methods in Natural Language Processing, pages 1–8, Hissar, Bulgaria, 2011.

[18] David Mareček and Zdeněk Žabokrtský. Exploiting reducibility in unsupervised dependency parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 297–307, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[19] David Mareček and Zdeněk Žabokrtský. Dealing with function words in unsupervised dependency parsing. In *Computational Linguistics and Intelligent Text Processing, CICLing 2014*, pages 250–261, Kathmandu, Nepal, 2014.

[20] Tahira Naseem, Harr Chen, Regina Barzilay, and Mark Johnson. Using universal linguistic knowledge to guide grammar induction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 1234–1244, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[21] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, 2016. European Language Resources Association.

[22] Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. The CoNLL 2007 Shared Task on Dependency Parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

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

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

[25] Yoav Seginer. Fast Unsupervised Incremental Parsing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 384–391, Prague, Czech Republic, 2007. Association for Computational Linguistics.

[26] Noah Ashton Smith. *Novel estimation methods for unsupervised discovery of latent structure in natural language text*. PhD thesis, Baltimore, MD, USA, 2007. AAI3240799.

[27] Anders Søgaard. From ranked words to dependency trees: two-stage unsupervised non-projective dependency parsing. In *Proceedings of TextGraphs-6: Graph-based Methods for Natural Language Processing*, TextGraphs-6, pages 60–68, Stroudsburg, PA, USA, 2011. Association for Com-

putational Linguistics.

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

[29] Valentin I. Spitkovsky, Hiyan Alshawi, Angel X. Chang, and Daniel Jurafsky. Unsupervised dependency parsing without gold part-of-speech tags. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, 2011.

[30] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. From baby steps to leapfrog: how "less is more" in unsupervised dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 751–759, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[31] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. Lateen EM: Unsupervised training with multiple objectives, applied to dependency grammar induction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, 2011.

[32] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. Punctuation: Making a point in unsupervised dependency parsing. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning (CoNLL-2011)*, 2011.

[33] Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. Three Dependency-and-Boundary Models for Grammar Induction. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012)*, 2012.

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

[35] Kewei Tu. Combining the sparsity and unambiguity biases for grammar induction. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 105–110, Montréal, Canada, June 2012. Association for Computational Linguistics.