Combining Supervised and Unsupervised Parsing for Distributional Similarity

Martin Riedl, Irina Alles and **Chris Biemann**
FG Language Technology
Computer Science Department, Technische Universität Darmstadt
Hochschulstrasse 10, D-64289 Darmstadt, Germany
{riedl,biem}@cs.tu-darmstadt.de, ialles@gmx.de

Abstract

In this paper, we address the role of syntactic parsing for distributional similarity. On the one hand, we are exploring distributional similarities as an extrinsic test bed for unsupervised parsers. On the other hand, we explore whether single unsupervised parsers, or their combination, can contribute to better distributional similarities, or even replace supervised parsing as a preprocessing step for word similarity. We evaluate distributional thesauri against manually created taxonomies both for English and German for five unsupervised parsers. While for English, a supervised parser is the best single parser in this evaluation, we find an unsupervised parser to work best for German. For both languages, we show significant improvements in word similarity when combining features from supervised and unsupervised parsers. To our knowledge, this is the first work where unsupervised parsers are systematically evaluated extrinsically in a semantic task, and the first work to show that unsupervised parsing can complement and even replace supervised parsing, when used as a pre-processing feature.

1 Introduction

While the field has seen increased interest in automatically inducing syntactic structures from raw or part-of-speech (POS) tagged text, the evaluation of unsupervised data-driven parsers has almost exclusively been conducted either by introspection or by automatic comparison to treebanks. It might be due to comparatively low scores on reproducing a treebank's syntactic annotation that hardly anyone has yet attempted to use the output of unsupervised parsers for an NLP task other than parsing itself.

A further complication with unsupervised parsers – be it dependency parsers, constituency parsers or combinatory categorial grammar parsers – is that the categories induced by such parsers cannot be straightforwardly mapped to linguistically-inspired categories as defined in a treebank. But also when considering only unlabeled syntactic annotations, an unsupervised parser is hardly to blame if it does not adhere to sometimes arbitrary conventions: e.g. for dependencies, it is not a priori clear how to connect auxiliary and main verbs, where to attach the complementizer of subordinate clauses, how to represent a conjunction and its conjuncts, how to relate the preposition and the nominal in prepositional phrases, and how to handle punctuation, cf. Nivre and Kübler (2006), Schwartz et al. (2011).

When it comes to *utilizing* syntactic structures, however, it is more important that they are consistent across different sentences than that they adhere to specific syntactic theories and conventions. Here, we choose a task that makes only intermediary use of syntactic structures: we employ unsupervised parsing for preprocessing corpora for the purpose of computing distributional similarities. Since it is generally accepted (e.g. (Lin, 1997; Curran and Moens, 2002)), that syntactic preprocessing plays an important role for the quality of distributional thesauri, and comparing words along their syntactic contexts does rely on the existence of such a structure rather than its actual representation, we believe that distributional similarities are an excellent test bed for addressing the following two research questions: (1) How do unsupervised parsers compare to supervised parsers when used as feature providers for building Dis-

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 1435–1446, Dublin, Ireland, August 23-29 2014.

tributional Thesauri (DTs) in comparison to supervised parsers? (2) Can the combination of syntactic parsers increase DT quality?

2 Related Work

2.1 Unsupervised Parser Evaluation

As with other unsupervised approaches, the premise of unsupervised induction of syntactic structure is to alleviate the bottleneck of expensive manual annotations for improving NLP applications. For grammar induction, the potential is extremely high due to the complexity of the subject matter: treebanks belong to the most work-intensive NLP datasets. On the other hand, this complexity is hard to grasp for unsupervised systems, which is probably the reason why unsupervised parsing technology is still in its infancy, despite more than a decade of work on this topic.

One of the early works inducing structure from raw sentences and yielding better performance than a random baseline was achieved by van Zaanen and of Leeds. School of Computer Studies (2001), who used an Alignment Based Learning (ABL) approach. This algorithm compares all sentences of a given set and considers matching sequences as constituents. Klein and Manning (2002) presented another approach focusing on constituent sequences called the Constituent-Context Model (CCM). It is an EM-based iterative approach that makes use of the linguistic phenomenon that long constituents often have shorter representations of the same grammatical function that occur in similar contexts. A hybrid approach combining CCM with a dependency model, called Dependency Model with Valence (DMV), shows even better performance and is the first unsupervised system to outperform the right-branching baseline (Klein and Manning, 2004). A great number of recent works are based on DMV, such as the system by Headden III et al. (2009), who improved DMV by adding lexical information, and Gillenwater et al. (2010) who added posterior regularization during the training process. Bod (2007) takes a slightly different direction by following an "all subtrees approach", where all possible binary trees are generated for each sentence. It generates all possible binary trees for each sentence. The parse of a new sentence is determined by selecting the most probable tree based on the previously accumulated subtree frequencies. Most of the evaluation of these parsers was performed against a treebank, offering manually annotated and linguistically motivated parse trees. Schwartz et al. (2011) underline the fact that treebanks contain linguistically problematic annotations, cases without linguistic consensus, such as the decision on the head of a verb phrase or a sequence of nouns. They show that the neglectance of these cases has a significant but unjustified negative influence on the evaluation outcomes and propose a new measure, Neutral Edge Direction (NED), which alleviates this problem. Bod (2007) argues that parser evaluation against a treebank favors supervised approaches and therefore measures the parser quality on the outcome of a syntax based Machine Translation (MT) task where the dependency parsers are evaluated as language models. In Motazedi et al. (2012), a single unsupervised parser is evaluated in an extrinsic evaluation for realisation ranking, and does not compare favorably against a supervised parser. Other extrinsic evaluations with supervised dependency parsers have been performed in information extraction systems (Miyao et al., 2008; Buyko and Hahn, 2010) or semantic role labeling (Johansson and Nugues, 2008).

2.2 Evaluating Distributional Similarity

Distributional thesauri have been evaluated both extrinsically and intrinsically. Extrinsic evaluations have been performed e.g. for automatic set expansion (Pantel et al., 2009) or phrase polarity identification (Goyal and Daumé, 2011). In this work, we will conduct an intrinsic evaluation, which is more common for the evaluation of DTs and lexical semantic similarity. Lin (1997; 1998) introduced two measures using WordNet (Miller, 1995) and Roget's Thesaurus. Using WordNet, he defines context (synsets a word occurs in Wordnet or subsets when using Roget's Thesaurus) and then builds a gold standard thesaurus using a similarity measure on these contexts. Then he evaluates his automatically computed Distributional Thesaurus (DT) with respect to the gold standard thesauri. Weeds et al. (2004) evaluate various similarity measures based on 1000 frequent and 1000 infrequent target terms. Curran (2004) created a gold standard thesaurus by manually extracting entries from several English thesauri for 70 words. His automatically generated DTs are evaluated against this gold standard thesaurus. All these

systems employ context representations based on syntactic parsing for computing word similarity.

We are going to use a comparatively simple WordNet-based measure, which calculates the similarity between two terms using the WordNet::Similarity path measure (Pedersen et al., 2004), and averages path scores between a target term and its n most similar terms. The score between two terms is inversely proportional to the shortest path between all the synsets of both terms. If two terms share a synset, the highest possible score of one is assigned. The score is 0.5 for terms that stand in a direct hypernym relation, and so on. While the absolute scores are hard to interpret due to inhomogeneity in the granularity of WordNet, they are well-suited for relative comparison when operating on the same set of target terms. The evaluation in this work is performed by comparing the average score of the top ten entries in the DT for each of the target terms and report separately on frequent and rare words. Riedl and Biemann (2013) also show that the results, using the WordNet based approach, are highly correlated to the results observed with Curran's approach using a manually created thesaurus. This justifies the usage of manually created taxonomies for this evaluation.

3 Methodology

3.1 Parsers

In our evaluation, we use five unsupervised parsers, which we will describe briefly. They have been selected to span several paradigms of unsupervised syntax induction, and due to software availability.

Gillenwater et al. (2010)[1] use a model based on the DMV (Klein and Manning, 2004) and improve performance by adding sparsity biases on dependency types. They assume a corpus annotated with POS tags. The aim of this bias is to limit unique head-dependent tag pairs, which is achieved by a constraint on model posteriors during the learning process.

The work of Marecek and Straka (2013)[2] is another enhancement of the DMV and is subsequently referred to as Unsupervised Dependency Parser (UDP). It additionally uses prior knowledge in the form of stop estimates that are computed on a large raw corpus using the reducibility principle: a sequence of words is considered as reducible if a word can be removed from the phrase and the remaining part appears another time in the corpus. The assumed property, that the first word of a reducible sequence does not have any left children and the last word of this sequence does not have any right children, is used for the calculation of such stop estimates. The authors show that estimates computed on a large corpus such as Wikipedia can be used for the parsing of new text.

Bisk and Hockenmaier (2013) use an EM approach to induce a Combinatory Categorial Grammar (CCG), based on very general linguistic assumptions. It creates a model that can be used to parse unseen data. The algorithm requires a corpus, previously assigned with POS tags, in order to be able to distinguish between word classes (mainly to find the verb), and employs general knowledge such as that sentences are headed by verbs. Further language-specific properties are induced from the training data.

Seginer (2007)[3] takes an incremental parsing and learning approach. It operates directly on the plain text without the need for POS tags, by using Common Cover Links (CCL), which can be directly converted to dependency arcs. This parser learns during parsing and can be used without a prior learning step. This should result in increased parsing quality towards later stages, which suggests several passes over the training data. The obtained model can then be reused to parse unseen data.

The approach of Søgaard (2012) is different from all other approaches discussed here: This algorithm does not require any training data and can operate with or without POS tags. For this reason, it can be applied to arbitrary amounts of data, since it operates sentence-wise without memorizing previous inputs, and produces non-projective dependency parses. The words of a phrase are ordered by centrality and a parse is determined by the ranking of a parsing algorithm, which uses general linguistic knowledge for grammar induction. This knowledge is inspired by the rules of Naseem et al. (2010), and the approach has been tuned (once and for all, for all languages) on development data from the Penn Treebank.

[1] http://code.google.com/p/pr-toolkit/
[2] http://ufal.mff.cuni.cz/udp/
[3] http://www.seggu.net/ccl/

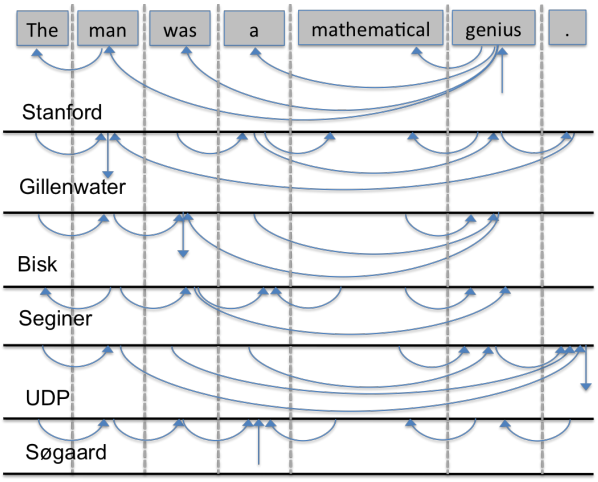

Figure 1: Parses for an example sentence for several parsers. Here, Bisk's parser looks most similar to the parses extracted from the Stanford parser. Gillenwater and UDP seem to have some problems with the full stop. Søgaards parser mostly connects neighbors.

	Baseline	Søgaard	Gillenwater	UDP	Bisk	Seginer	Seginer
English	53.2	59.9	64.4	55.4	70.3	55.6 (WSJ 40)	74.2 (WSJ 10)
German	33.7	57.6	35.7	52.4	68.4	38.2 (Negra 40)	48.0 (Negra 10)

Table 1: Unlabeled accuracy values of different unsupervised parsers based on the CoNLL-X shared task (Buchholz and Marsi, 2006). Seginers results show F-measure values for the Negra and the WSJ corpus, used with maximum sentence of lengths of 10 and 40.

An example sentence and the according parses coming from the 10M model, except for UDP, where the 1M model is used (cf. Table 2 in Section 4.3.1), are shown in Figure 1.

Table 1 reports the accuracy of four parsers for the English and the German treebanks from the CoNLL-X shared task (Buchholz and Marsi, 2006) predicting unlabeled dependency parses for sentences with length equal and smaller than 10 tokens. Seginer reports only F-scores for WSJ and Negra considering sentences with a maximum length of 10 and 40. The best baselines reported in Canisius et al. (2006) are a left branching method for English and a nearest neighbor branching method for German, which is a combination of left and right branching.

3.2 Computing Distributional Thesauri

The extraction of context features, used to calculate similarities between terms, is performed in accordance with the generic scheme proposed in (Biemann and Riedl, 2013): A (typed or untyped) parser arc is split into term and context feature, which consists of the edge direction and label (if any), and the connected term. Similarity between terms is subsequently computed on the overlap of their most salient context features. We represent the term t and the context feature c as a pair $< t, c >$ and extract a dependency triple (or dependency pair, as most unsupervised dependency parsers do not label the edges). For the dependency between *I* and *gave* `(nsub;gave;I)` in *I gave her the book*, terms and context features would look like `<gave,(nsub,I,@)>` and `<I,(nsub,@,gave)>`. In this example, the term *gave* is characterized by the context information that *I* is its nominal subject, and term *I* is characterized by being the subject of *gave*. We build distributional thesauri using the JoBimText[4] open-source framework. This framework scales to large data and has proven to outperform other methods, when using large data (Riedl and Biemann, 2013). The computation of the distributional thesaurus within this framework is following the MapReduce paradigm and scales to very large corpora. This is achieved by applying a significance measure between term and context feature, retaining only the most salient 1000 context features per term, and computing the cardinality of the set overlap between the respective context features

[4] `www.jobimtext.org`, (Biemann and Riedl, 2013)

per term, which defines the similarity between terms. Per term, the most similar terms are subsequently ranked, resulting in a distributional thesaurus as introduced by Lin (1997).

4 Evaluation

We report experimental results on German and English corpora. Both corpora are compiled from 10 million sentences (about 2 Gigawords) each from the Leipzig Corpora Collection[5], randomly sampled from online newspapers. The semantic similarity in English DTs is assessed using WordNet 3.1 as a lexical resource, as proposed by Riedl and Biemann (2013). For evaluating the German DTs, we replace WordNet by its German counterpart, GermaNet 8 (Hamp and Feldweg, 1997). We report results separately for frequent and infrequent targets and average the path scores for the most similar 10 words per entry. The evaluation of the English DTs is performed using 1000 frequent and 1000 infrequent nouns, as previously employed by Weeds et al. (2004). These nouns are randomly sampled from the British National Corpus (BNC) and all occur in WordNet. For the evaluation of German DTs, we randomly selected 1000 frequent and 1000 infrequent nouns from our German corpus that all occur in GermaNet.

4.1 Experimental Settings

The DTs are calculated using the dependencies from the unsupervised parsers, one at a time. To show the impact of corpus size, we down-sampled our corpora, and used 1 million (1M), 100,000 (100K) and 10,000 (10K) sentences (raw or automatically POS-tagged with the TreeTagger[6]) for training/inducing the parsers. Not all parsers were able to deal with the large training sets in feasible runtime, which might either be due to their computational complexity or their implementation. While it would be preferable to keep the corpus size for DT computations constant, this was not possible since some of our unsupervised parsers cannot be applied to unseen text. Hence, we decided to report DT quality results for two setups: Setup A uses the same data for training the parsers and the DT computation. Setup B uses the full corpus of 10M sentences for DT computation, parsed with unsupervised parsers induced on differently sized corpus samples. We feel that Setup B is better reflecting the possible utilization of unsupervised parsers for semantic similarity, since DT quality is known to increase with corpus size. However, we still wanted to assess the quality of parsers that cannot be operated on unseen text in their current state of development.

4.2 Four Baselines

We compare the results of unsupervised parsers against four baselines. As a lower-bound baseline, we use a random dependency parser that connects each word in a sentence with a randomly chosen other word. As a supervised upper-bound baseline, we use Stanford collapsed dependencies (Marneffe et al., 2006) for the English data and dependencies coming from the Mate tools (Bohnet, 2010) for the German corpus. Finally, to gauge whether the potential of unsupervised parsing to model long-range dependencies – as opposed to local n-gram contexts – lead to better distributional similarities, we use word bigrams and trigrams as n-gram-based systems. The bigram system simulates left- and right-branching. We characterize the word in the first and in the second position of two neighboring words, which results to the following term feature pairs according to the example in Section 3.2: `<I,(@,gave)>` and `<gave,(I,@)>`. Using the trigram, we characterize the word in the second position with the context feature formed by the pair of words in first and third position. The term-feature pair for *gave* would be `<gave, (I,@,her)>`.

While we expect the scores of any reasonable unsupervised parser to fall somewhere between the lower bound and the upper bound when compared in the same setting, the n-gram baselines serve as a measure for whether it is worth the trouble to induce and run the unsupervised parser for our evaluation application, as opposed to relying on an arguably simpler system for this purpose.

[5] `corpora.uni-leipzig.de`, (Richter et al., 2006)

[6] `www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`, (Schmid, 1997)

4.3 Results

4.3.1 Single Parser Results for English

We summarize the results for the English evaluation for Setup A and Setup B in Table 2. All unsupervised parsers beat the random baseline in all setups, with higher improvements observed using more training data, which somewhat validates their approaches. Also, more data for DT computation results in higher similarity scores, and rare words generally receive lower scores on average, which is expected and validates the DT computation framework.

		10k		100k		1M		10M	
	Parser	freq	rare	freq	rare	freq	rare	freq	rare
Setup A	Random	0.115	0.029	0.128	0.082	0.145	0.103	0.159	0.113
	Trigram	0.133	0.020	0.179	0.082	0.200	0.120	0.236	0.151
	Bigram	0.140	0.029	0.173	0.088	0.208	0.129	0.246	0.159
	Stanford	0.151	0.028	0.209	0.128	0.261	0.176	0.280	0.209
	Seginer	0.136	0.027	0.176	0.085	**0.211**	0.127	0.240	0.155
	Gillenwater	0.135	0.026	0.159	0.077	0.195	0.117	0.223	0.141
	Søgaard	0.120	0.027	0.147	0.083	0.185	0.117	0.227	0.144
	UDP	0.127	0.017	0.169	0.063	0.204	0.119	*	*
	Bisk	0.118	0.017	*	*	*	*	*	*
Setup B	Seginer	0.200	0.063	0.236	0.139	0.241	0.156	0.240	0.155
	Gillenwater	0.220	0.140	0.221	0.142	0.221	0.141	0.223	0.141
	Søgaard	0.227	0.144	0.227	0.144	0.227	0.144	0.227	0.144
	Bisk	0.220	0.139	*	*	*	*	*	*

Table 2: Setup A English: Parser induction and DT computation on the same corpus. Wordnet path scores averaged on top 10 similar words, for 1000 frequent and 1000 rare nouns. A * denotes that the evaluation failed because of computational constraints. Setup B English: Parser induction on different corpus sizes, and DT computation on 10M sentences.

In comparison to the n-gram baselines, only the parser by Seginer yields a higher score for frequent words and 1M sentences training in Setup A. However, the difference is very small and is confirmed on the 10M sentences only in comparison to the Trigram baseline. It seems that Seginer's training procedure saturates somewhere between 100K and 1M sentences, and shows even slightly worse performance on 10M sentences of training in Setup B. All parsers do not seem to be particularly useful as preprocessing steps for DT computation, since better similarity can consistently be reached by context features based on bigram statistics.

Comparing the unsupervised parsers, we note that Seginer's approach consistently scores highest in Setup A, while UDP comes in second for frequent words but not for rare words. While Gillenwater's approach reaches comparably high scores for small corpora in Setup A, it is beaten by Søgaard's no-training approach for larger corpora: It seems that Gillenwater's training procedure can hardly make use of additional training, which is shown in Setup B, where practically no differences are observed between 10K and 10M sentences of parser training. Differences in Setup A are thus solely due to increased corpus size for DT computation for the Gillenwater experiments.

UDP did not finish parsing 10M sentences after running for 157 days, and it is not trivial to disable its update procedure, which is why we could not include UDP in Setup B. Bisk's parser is a special case in this evaluation, since it only selects sentences shorter than 15 tokens for training, and hence was effectively trained on a 5400 sentence subset of the 10K corpus. While we did not manage to train it on larger corpora, we could apply this model on 10M sentences in Setup B, where it lands slightly below the no-training Søgaard parser, but clearly above Seginer's approach for 10K training.

4.3.2 Single Parser Results for German

A different picture is drawn for the German evaluation (see Setup A in Table 3). Comparing the results of the unsupervised parsers, Seginer's parser does not only outperform the trigram and bigram baseline for frequent nouns but also the supervised Mate parser for all corpus sizes. Yet, the improvements over

the Mate parser are not significant for all results using a paired t-test[7]. Also, Søgaards parser exceeds the trigram and bigram baseline for 10 million sentences. The remaining unsupervised parsers can beat the random baseline for frequent nouns but none of the n-gram baselines. Again we are not able to parse the 10 million sentences using UDP and also Gillenwater's parser failed, parsing this corpus. Comparing the baselines in Setup A (see Table 3), we observe a difference between the sophisticated baselines and the random baseline only for frequent words.

		10k		100k		1M		10M	
	Parser	freq	rare	freq	rare	freq	rare	freq	rare
Setup A	Random	0.097	0.002	0.108	0.010	0.123	0.051	0.143	0.077
	Trigram	0.102	0.002	0.130	0.014	0.159	0.067	0.179	0.086
	Bigram	0.112	0.003	0.130	0.009	0.163	0.053	0.192	0.082
	Mate	0.111	0.004	0.126	0.014	0.170	0.027	0.204	0.090
	Seginer	**0.113**†	0.002	**0.137**†	0.012	**0.171**	0.068	**0.208**	0.091
	Gillenwater	0.104	0.002	0.118	0.009	0.132	0.040	*	*
	Søgaard	0.104	0.002	0.123	0.010	0.161	0.054	0.193	0.077
	UDP	0.107	0.001	0.129	0.004	0.151	0.021	*	*
	Bisk	0.101	0.002	*	*	*	*	*	*
Setup B	Seginer	0.153	0.004	0.186	0.021	0.200	0.092	0.208	0.091
	Gillenwater	0.189	0.080	0.190	0.082	0.189	0.080	*	*
	Søgaard	0.193	0.077	0.193	0.077	0.193	0.077	0.193	0.077
	Bisk	0.185	0.069	*	*	*	*	*	*

Table 3: Setup A and B for German corpora.

Furthermore, we see that the supervised Mate parser results in worse scores for the frequent nouns using the 10k and 100k dataset in comparison to the bigram baseline. This could be attributed to the heavier tail in German's word frequency distribution, which results in sparser context features for small data[8]. For the 1M and 10M datasets, the supervised parser yields the best similarities for frequent nouns.

The results for Setup B for the German corpora are shown at the bottom in Table 3. We observe similar trends to the ones for the English data: using more data for the training does not seem to help the performance of Gillenwater's algorithm. Noticeable is the increase of Seginer's results for rare words as training data size increases. Seginer's algorithm even beats both n-gram baselines for the 10M corpus when trained only on 1 million sentences.

4.3.3 Combining Parsers for DT Quality Improvement

To clarify the best practice for building a DT of high quality, we combine different parsers: the two best-performing unsupervised parsers (Søgaard's and Seginer's), the baselines and the supervised parser. Additionally, these two parsers where the only ones which could be applied to the largest dataset for both languages.

For English (see Table 4), we observe a boost in performance when combining unsupervised parsers. Combining the supervised Stanford parser with the bigram and the trigram baselines also leads to a significant improvement ($p < 0.01$)[9] over the Stanford parser alone, which is about the same as combining the supervised parser with the two unsupervised parsers, and combining all five types of features for DT construction. Overall, a relative improvement of 3.5% on the average WordNet::Path measure for frequent words and a relative 4% improvement for rare words is obtained over the Stanford parser alone.

The results for German (see Table 5) show a similar trend. It is remarkable that merging the two unsupervised parsers already outperforms the supervised Mate parser significantly[9] with $p < 0.01$ (6.7% for frequent and 8% relative improvement for rare words). The combination of the supervised and unsupervised parsers again leads to further improvement, which is also significant over the supervised parser alone, and again, adding the bigram and trigram baselines to the three parsers does not help.

[7]Significant improvements ($p < 0.01$) against the Mate parser are marked with the symbol † in Table 3 for frequent nouns.

[8]Within the 10M sentences, there are 22 million word types in the German corpus and 10 million word types in the English corpus.

[9]We use a paired t-test to compare the DTs built using the supervised parser and the combinations.

Parser	frequent	rare
Stanford (supervised)	0.280	0.209
Seginer	0.240	0.155
Søgaard	0.227	0.144
Seginer & Søgaard	0.248	0.162
Stanford & Bigram & Trigram	**0.290**†	**0.217**†
Stanford & Seginer & Søgaard	**0.291**†	**0.217**†
Stanford & Seginer & Søgaard & Bigram & Trigram	**0.290**†	**0.218**†

Table 4: Combinations of different parsers for computing English thesauri. The cross (†) indicates significant improvements over the supervised parser.

Parser	frequent	rare
Mate (supervised)	0.204	0.090
Seginer	0.208	0.091
Søgaard	0.193	0.077
Seginer & Søgaard	0.218†	0.097†
Mate & Bigram & Trigram	0.204	0.091
Mate & Seginer & Søgaard	**0.222**†	**0.100**†
Mate & Seginer & Søgaard & Bigram & Trigram	**0.222**†	**0.100**†

Table 5: Combinations of different parsers for computing German thesauri

4.3.4 Discussion

Overall, it is surprising how well Søgaard's parser performs in comparison to others on this task, since it neither uses training nor relies on POS tags. This hints at either unsupervised parsing being simpler than commonly assumed or rather the immaturity of all unsupervised parsers tested. Further, we would have expected that trained unsupervised parsers, as most unsupervised methods, would exhibit a better performance when trained on larger corpora. This could not be confirmed for both systems that we trained on various corpus sizes, i.e. Seginer's and Gillenwater's approach. The findings are only moderately correlated with evaluations on treebanks, cf Table 1: Whereas Seginer's and Søgaard's parsers perform favorably in our evaluation, they are outperformed by Bisk's parser on treebanks, which currently does not scale to large data. Gillenwater's parser seems to be overly tuned to English treebanks, but cannot capitalize on this in our evaluation for English.

POS information does not seem beneficial for unsupervised parser induction in noun similarity evaluation, since the highest-scoring approach by Seginer does not use POS tags and a version of Søgaard's parser with POS tags scored slightly but consistently lower than the version without POS, as we found in further experiments. This is in line with the findings of Cramer (2007), who reports no benefit from manually corrected or unsupervised POS tags for a range of unsupervised parsers.

Comparing the results of previous intrinsic evaluations (see Table 1) and the results of our extrinsic evaluation (see Table 2 and 3), we observe that the ranking of parsers is only mildly correlated. Thus, our proposed evaluation covers different aspects than the adherence to (partially arbitrary) conventions of manually labeled dependency data. Also, our current evaluation disregards all arcs that do not involve nouns.

When combining parsers, we observe that we can improve the quality of DTs significantly. This leads us to conclude that unsupervised parsers should at least be used for generating features when computing distributional thesauri of high quality. In case no high-quality supervised parser is available for the language or domain of interest, it might suffice to use combinations of unsupervised parsers.

We also report the computation times of the different parsers, for the English dataset for Setup A (see Table 6). The results were computed on a server with 80 GB and 16 cores. Whereas all parsers require different amounts of memory, all parsers are single-threaded[10]. While Søgaard's parser is the fastest for small datasets, Seginer's runs faster on 10 million sentences. Whereas Gillenwater's and Seginer's

	10k	100k	1M	10M
Seginer	210	224	261	**508**
Gillenwater	3248	3248	3280	5546
Søgaard	**3**	**21**	**182**	975
UDP	183	1220	11316	-

Table 6: Computation time in minutes for parsing the data according to the English corpora used in Setup A, cf. Table 2

[10] As Søgaards algorithm parses sentence-wise without storing any information, it could be easily run multi-threaded.

algorithm require almost the same time for parsing 10k, 100k or 1M sentences, the runtime of the UDP and Søgaard's parser is linear in time with the number of sentences to be parsed. We cannot report the parsing times for the Bisk algorithm, as the parsing was not performed by us. Again it is noticeable that the best two parsers are also the two unsupervised parsers that run quickest.

5 Conclusion

The contribution of this paper is two-fold: First, we have proposed and conducted a comparative extrinsic evaluation procedure for unsupervised parsers based on noun similarity in DTs. Second, we have explored how to improve DT quality by combining features from several parsers. The transparency of this method with respect to the kind of induced structures (dependencies, constituent trees, combinatory categorial grammar) and with respect to labels of nodes and edges in the parse graph makes it possible to compare different unsupervised parsers without having to rely on treebanks. Since semantic similarity, especially for nouns, is a building block for many NLP applications, we feel that removing the dependency on high-quality supervised parsers can give rise to semantic technologies in many languages. We have conducted this evaluation with five different unsupervised parsers, and examined the influence of corpus size for parser training and for the similarity computation in a series of experiments. Using established methods for evaluating distributional similarity against lexical semantic resources, we were able to measure differences between parsers in this task that are not reflected by intrinsic evaluations that compare their induced structures to treebanks. These include the influence of corpus size on the training procedure and the consistency of parse fragments on "frequent versus rare words" as well as different languages. Further, we were able to pinpoint a crucial point in unsupervised parsers that has not received much attention: approaches that do not induce an actual parser that can be run on unseen sentences but merely produce syntactic annotations for a given fixed training corpus are hardly useful in applications.

Our evaluation results can be summarized as follows: For English, with its relatively fixed order, Seginer's parser achieves very scarce to no improvements compared to a simple n-gram baseline when used to compute distributional similarities. But for German, Seginer's parser outperforms all baselines including a state-of-the-art supervised parser, and Søgaard's simplistic approach compares favorably to the n-gram baselines. Furthermore, we demonstrate that the quality of noun similarity can be improved significantly when combining the features from supervised and unsupervised parsers.

While today's unsupervised parsers might not be ready for their utilization for semantic similarity for the English language, they can be applied to a large number of other languages where highly optimized supervised parsers are not available. Additionally, we feel that our proposed evaluation method exhibits enough sensitivity to be a meaningful test bed for future unsupervised parsers.

6 Outlook

Where do we go from here? We strongly argue that in times of availability of very large monolingual corpora from the web, we should strive at unsupervised parser induction systems that can make use of large training data, as opposed to focussing our efforts on complex models that scale poorly, and thus cannot elevate to the performance levels needed in order to make unsupervised parsing a building block in natural language processing applications.

For further work, we want to proceed in several ways: we would like to extend our evaluation framework from nouns to other parts of speech. Furthermore, we will explore whether unsupervised parsers can be tuned towards the task of computing a distributional thesaurus, e.g. by using only assignments with a certain confidence, type, or from sentences with limited length. Additionally, we would like to explore the interaction of unsupervised POS induction and grammar induction (Headden, III et al., 2008), in order to entirely remove language-dependent preprocessing for the purpose of semantic similarity computations, while at the same time being able to leverage the advantages of structured representations, cf. Erk and Padó (2008). Finally, we would like to test whether we can also detect a different ranking for different supervised parsers when comparing their scores in the normal treebank setting versus using them for building distributional thesauri.

Acknowledgments

This work has been supported by the German Federal Ministry of Education and Research (BMBF) within the context of the Software Campus project *LiCoRes* under grant No. 01IS12054, by IBM under a Shared University Research Grant and by DFG under the *SemSch* project grant.

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.

Yonatan Bisk and Julia Hockenmaier. 2013. An HDP Model for Inducing Combinatory Categorial Grammars. In *Transactions of the Association for Computational Linguistics*, pages 75–88, Atlanta, GA, USA.

Rens Bod. 2007. Is the end of supervised parsing in sight? In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 400–407, Prague, Czech Republic.

Bernd Bohnet. 2010. Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 89–97, Beijing, China.

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, New York City, New York.

Ekaterina Buyko and Udo Hahn. 2010. Evaluating the impact of alternative dependency graph encodings on solving event extraction tasks. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 982–992, Cambridge, Massachusetts.

Sander Canisius, Toine Bogers, Antal van den Bosch, Jeroen Geertzen, and Erik Tjong Kim Sang. 2006. Dependency parsing by inference over high-recall dependency predictions. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06, pages 176–180, New York City, New York.

Bart Cramer. 2007. Limitations of Current Grammar Induction Algorithms. In *Proceedings of the ACL 2007 Student Research Workshop*, pages 43–48, Prague, Czech Republic.

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 '02, pages 59–66, Philadelphia, Pennsylvania, USA.

James R. Curran. 2004. *From Distributional to Semantic Similarity*. Ph.D. thesis, University of Edinburgh.

Katrin Erk and Sebastian Padó. 2008. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 897–906, Honolulu, Hawaii.

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

Amit Goyal and Hal Daumé, III. 2011. Generating semantic orientation lexicon using large data and thesaurus. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, WASSA '11, pages 37–43, Portland, Oregon, USA.

Birgit Hamp and Helmut Feldweg. 1997. GermaNet - a Lexical-Semantic Net for German. In *In Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15, Madrid, Spain.

William P. Headden, III, David McClosky, and Eugene Charniak. 2008. Evaluating unsupervised part-of-speech tagging for grammar induction. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 329–336, Manchester, United Kingdom.

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*, pages 101–109, Boulder, CO, USA.

Richard Johansson and Pierre Nugues. 2008. The effect of syntactic representation on semantic role labeling. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 393–400, Manchester, United Kingdom.

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*, 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*, pages 478–485, Barcelona, Spain.

Dekang Lin. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, ACL '98, pages 64–71, Madrid, Spain.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics - Volume 2*, COLING '98, pages 768–774, Montreal, Quebec, Canada.

David Marecek and Milan Straka. 2013. 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*, pages 281–290, Sofia, Bulgaria.

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.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38:39–41.

Yusuke Miyao, Rune Stre, Kenji Sagae, Takuya Matsuzaki, and Jun'ichi Tsujii. 2008. Task-oriented evaluation of syntactic parsers and their representations. In *Proceeding of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 46–54, Columbus, Ohio.

Yasaman Motazedi, Mark Dras, and François Lareau. 2012. Is bad structure better than no structure?: Unsupervised parsing for realisation ranking. In *Proceedings of the 24th International Conference on Computational Linguistics*, COLING '12, pages 1811–1830, Mumbai,India.

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*, pages 1234–1244, Cambridge, MA, USA.

Joakim Nivre and Sandra Kübler. 2006. Dependency parsing. In *Tutorial at COLING-ACL*, Sydney, Australia.

Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. 2009. Web-scale distributional similarity and entity set expansion. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 938–947, Singapore.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. WordNet::Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, HLT-NAACL–Demonstrations '04, pages 38–41, Boston, Massachusetts, USA.

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

Martin Riedl and Chris Biemann. 2013. Scaling to large^3 data: An efficient and effective method to compute distributional thesauri. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013*, pages 884–890, Seattle, WA, USA.

Helmut Schmid. 1997. Probabilistic part-of-speech tagging using decision trees. In Daniel Jones and Harold Somers, editors, *New Methods in Language Processing*, Studies in Computational Linguistics, pages 154–164. UCL Press, London, GB.

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

Yoav Seginer. 2007. Fast unsupervised incremental parsing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 384–391, Prague, Czech Republic.

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

Menno van Zaanen and University of Leeds. School of Computer Studies. 2001. *Building Treebanks Using a Grammar Induction System*. Research report series. University of Leeds, 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 '04, pages 1015–1021, Geneva, Switzerland.