

Improvements in unsupervised co-occurrence based parsing

Christian Hänig

Daimler AG

Research and Technology

89081 Ulm, Germany

christian.haenig@daimler.com

Abstract

This paper presents an algorithm for unsupervised co-occurrence based parsing that improves and extends existing approaches. The proposed algorithm induces a context-free grammar of the language in question in an iterative manner. The resulting structure of a sentence will be given as a hierarchical arrangement of constituents. Although this algorithm does not use any a priori knowledge about the language, it is able to detect heads, modifiers and a phrase type's different compound composition possibilities. For evaluation purposes, the algorithm is applied to manually annotated part-of-speech tags (POS tags) as well as to word classes induced by an unsupervised part-of-speech tagger.

1 Introduction

With the growing amount of textual data available in the Internet, unsupervised methods for natural language processing gain a considerable amount of interest. Due to the very special usage of language, supervised methods trained on high quality corpora (e. g. containing newspaper texts) do not achieve comparable accuracy when being applied to data from fora or blogs. Huge annotated corpora consisting of sentences extracted from the Internet barely exist until now.

Consequently a lot of effort has been put into unsupervised grammar induction during the last years and results and performance of unsupervised parsers improved steadily. Klein and Manning (2002)'s constituent context model (CCM) obtains 51.2% f-score on ATIS part-of-speech strings. The same model achieves 71.1% on Wall Street Journal corpus sentences with length of at most 10 POS tags. In (Klein and Manning, 2004) an approach combining constituency and

dependency models yields 77.6% f-score. Bod (2006)'s all-subtree approach — known as Data-Oriented Parsing (DOP) — reports 82.9% for UML-DOP. Seginer (2007)'s common cover links model (CCL) does not need any prior tagging and is applied on word strings directly. The f-score for English is 75.9%, and for German (NEGRA10) 59% is achieved. Hänig et al. (2008) present a co-occurrence based constituent detection algorithm which is applied to word forms, too (unsupervised POS tags are induced using unsuPOS, see (Biemann, 2006)). An f-score of 63.4% is reported for German data.

In this paper, we want to present a new unsupervised co-occurrence based grammar induction model based on Hänig et al. (2008). In the following section, we give a short introduction to the base algorithm *unsuParse*. Afterwards, we present improvements to this algorithm. In the final section, we evaluate the proposed model against existing ones and discuss the results.

2 Co-occurrence based parsing

It has been shown in (Hänig et al., 2008) that statistical methods like calculating significant co-occurrences and context clustering are applicable to grammar induction from raw text. The underlying assumption states that each word prefers a certain position within a phrase. Two particular cases are of special interest: a word's occurrence at the beginning of a sentence and a word's occurrence at the end of a sentence. Those positions obviously are constituent borders and can be easily used to extract syntactic knowledge. One possibility is to discover constituents employing constituency tests (see (Adger, 2003)), whereby these two cases can be used to express and use one of them in a formal way: the movement test.

Three neighbourhood co-occurrences express the aforementioned observations:

- Value a denotes the significance of word A standing at the last position of a sentence (where $\$$ is an imaginary word to mark a sentences' end).

$$a = sig(A, \$) \quad (1)$$

- Contrary, variable b denotes the significance of a word B being observed at the beginning of a sentence (where $\hat{\cdot}$ is an imaginary word to mark the beginning of a sentence).

$$b = sig(\hat{\cdot}, B) \quad (2)$$

- Additionally, a third value is necessary to represent the statistical significance of the neighbourhood co-occurrence containing word A and B .

$$c = sig(A, B) \quad (3)$$

To compute those significance values for a corpus, the log-likelihood measure (see (Dunning, 1993)) is applied using corpus size n , term frequencies n_A and n_B (for the words A and B) and frequency n_{AB} of the co-occurrence of A and B .

To detect constituent borders between two words, a separation value sep_{AB} can be defined as:

$$sep_{AB} = \frac{a}{c} \cdot \frac{b}{c} = \frac{a \cdot b}{c^2} \quad (4)$$

If word A occurs more significantly at the end of a sentence as in front of B , then $\frac{a}{c} > 1$. Additionally, b is larger than c if B is observed more significantly at the beginning of a sentence as after A and $\frac{b}{c}$ will be > 1 . In this case sep_{AB} is > 1 and obviously, a constituent border would be situated between A and B .

The basic approach to create parse trees from separation values between two adjacent words is to consecutively merge the two subtrees containing the words with the smallest separation value between them — starting with each word in a separate subtree. In order to avoid data sparseness problems, co-occurrences and separation values are primarily calculated on part-of-speech tags. However, word co-occurrences will be used to preserve word form specific dependencies.

In this paper, we want to present *unsuParse+* — an extension of this co-occurrence based approach. The first extension is the distinction between endocentric and exocentric elements which introduces the detection of heads along with their

modifiers (see section 2.2). Furthermore, learning of recursive constructions is facilitated. Secondly, we will consider discontinuous dependencies and present a possibility to detect rare constructions like complex noun phrases (see section 2.3). As third enhancement, we employ a simple clustering algorithm to induced phrases in order to detect constituents holding identical syntactic functions. Those phrases will be labeled the same way instead of by different phrase numbers (see section 2.4).

First, we will start with the detection of constituent candidates.

2.1 Detection of constituent borders

Instead of using sep_{AB} to detect constituent borders we use neighbourhood co-occurrence significances on account of an experiment in (Hänig et al., 2008) showing that the pure significance value c is sufficient.

Furthermore, we do not restrict the detection of phrases to bigrams and allow the detection of arbitrary n-grams. The motivation behind this is basically caused by coordinating conjunctions for which discussions on the *correct*¹ structure are raised. While Chomsky (1965) argues in favor of symmetric multiple-branching coordinating constructions (see Figure 1), recent discussions in the context of unification grammars (especially head-driven phrase structure grammar (see (Pollard and Sag, 1994)) prefer asymmetric endocentric constructions (see (Kayne, 1995) and (Sag, 2002)). The corresponding structure can be seen in Figure 2. Nevertheless, a symmetric construction containing two heads seems to be more appropriate for some languages (e. g. German, see (Lang, 2002)).

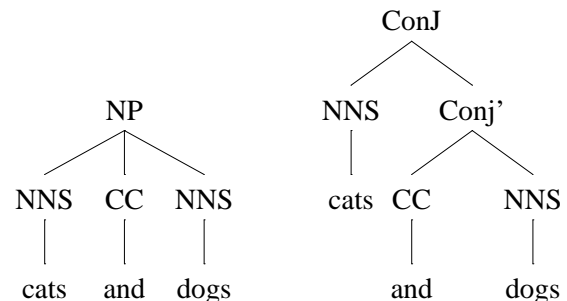


Figure 1: Symmetric coordinating conjunction

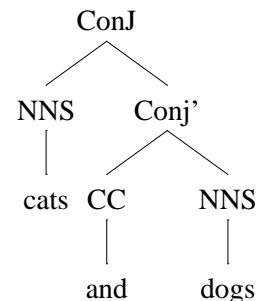


Figure 2: Asymmetric coordinating conjunction

¹correct meaning considered to be correct

Thus, the presented algorithm is able to deal with phrases containing any number of compounds.

As in (Hänig et al., 2008), phrases will be learned in an iterative manner (see details in section 2.5). Within each iteration, the n-gram P yielding the highest significance is considered to be the best candidate for being a valid constituent.

$$P = [p_0 \dots p_{n-1}] \quad (5)$$

The preferred position of part-of-speech tags is maintained as we define $pref(A)$ for every POS tag A . This value is initialized as the ratio of two particular significances as in Equ. 6:

$$pref(A) = \frac{sig(\hat{\cdot}, A)}{sig(A, \$)} \quad (6)$$

Analogous to sep_{AB} (see section 2) $pref(A) > 1$ if POS tag A prefers the first position within a phrase and vice versa.

Before a phrase candidate is used to create a new grammar rule, its validity has to be checked. Using the assumption that every word prefers a certain position within a constituent leads us to check the first word of a phrase candidate for preferring the first position and the last word for favoring the last one.

But there are at least two exceptions: coordinating conjunctions and compound nouns. Those constructions (e. g. *cats/NNS and/CC dogs/NNS, dog/NN house/NN*) usually start and end with the same phrase respectively POS tag. This would lead to wrong validation results, because NNS or NN do prefer the last position within a constituent and should not occur at the beginning. As both constructions are endocentric, they prefer the head’s position within the superordinating phrase and thus, their existence does not stand in contrast to the assumption made about preferred positions.

Formally, we get the following proposition:

$$\begin{aligned} valid(P) \Leftrightarrow & \quad p_0 = p_{n-1} \quad \vee \\ & \quad pref(p_0) \geq \varphi \wedge pref(p_{n-1}) \leq \frac{1}{\varphi} \end{aligned} \quad (7)$$

An *uncertainty* factor is introduced by φ , as some parts-of-speech tend to not appear at the borders of a sentence although they prefer a certain position within constituents. Some examples (given in Table 1) of the 5 most frequent English² and Ger-

man³ parts-of-speech will demonstrate this effect.

English		German	
NN	0.08	NN	0.30
IN	31.45	ART	242.48
NNP	1.39	APPR	143.62
DT	84.19	ADJA	5.06
NNS	0.31	NE	1.11

Table 1: Values of $pref(POS)$ for the 5 most frequent parts-of-speech of English and German

In both languages proper nouns (*NNP* resp. *NE*) occur slightly more often at the beginning of a sentence than at its end, although proper nouns prefer — like normal nouns — the last position of a phrase. To account for this effect, $pref(A)$ will be iteratively adapted to the observations of learned grammar rules as given in Equ. 8:

$$\begin{aligned} pref(p_0) &\leftarrow \frac{1}{\delta} \cdot pref(p_0) \\ pref(p_{n-1}) &\leftarrow \delta \cdot pref(p_{n-1}) \end{aligned} \quad (8)$$

Due to iterative learning of rules, we can use knowledge obtained during a previous iteration. Every rule contains reinforcing information about the preferred position of a part-of-speech. $pref(A)$ is adapted by a factor δ (with $0 < \delta < 1$) for the corresponding parts-of-speech and it will converge to its preferred position.

In later iterations, significances of phrase candidates do not differ considerably from each other and thus, the order of phrase candidates is not very reliable anymore. Consequently, parts-of-speech occur at non-preferred positions more often and trustworthy knowledge (in form of $pref(A)$) about the preferred positions of parts-of-speech is very helpful to avoid those phrase candidates from being validated.

We want to give one example for English: adjectives (*JJ*). Before the first iteration, $pref(JJ)$ is initialized with 1.046 which means that *JJ* has no preferred position. The most significant rules containing *JJ* are *JJ NN*, *JJ NNS* and *JJ NNP* — supporting a preference of the first position within a constituent. An iterative adaption of $pref(JJ)$ will represent this observation and disapprove constituents ending with *JJ* (like *DT JJ* or *IN JJ*) in upcoming iterations.

²Penn Tree Tagset, see (Marcus et al., 1993)

³Stuttgart-Tübingen Tagset, see (Thielen et al., 1999)

After having detected a new and valid constituent, we can use context similarity and other statistical methods to learn more about its behaviour and inner construction.

2.2 Classification into endocentric and exocentric constructions

Endocentric constructions contain a head — or more than one in symmetric coordinate constructions — which is syntactically identical to the endocentric compound. Additionally, at least one optional element subordinating to the head is contained in the construction. An exocentric construction on the other hand does not contain any head element which is syntactically identical to the whole construction.

The following example sentences will demonstrate the distinction of these two types. Sentence (a) contains a determiner phrase (DP: *a new car*) which has a noun phrase embedded (NP: *new car*). The NP can be replaced by its head as in sentence (b) and thus is regarded to be endocentric. The DP is exocentric — it can neither be replaced by the determiner (sentence (c)) nor by the NP (sentence (d)) without losing its syntactical correctness.

- (a) I buy a new car.
- (b) I buy a car.
- (c) * I buy a.
- (d) * I buy new car.

Detection of endocentric constructions yields valuable information about the language in question. It is possible to detect heads along with their modifiers without any a priori knowledge. Furthermore, detection of optional modifiers reduces the complexity of sentences and thus, facilitates learning of high precision rules.

Without classification into endocentric and exocentric constructions, two rules ($P\#1 \leftarrow JJ\ NN$ and $P\#2 \leftarrow JJ\ P\#1$) would be necessary to parse the phrase *first civil settlement* as given in Figure 3. Using knowledge about subordinating elements achieves the same result (see Figure 4) with one rule ($NN \leftarrow JJ\ NN$). Additionally, data-sparseness problems are circumvented as no rare occurrences like $JJ \dots JJ\ NN$ need to be contained in the training corpus to eventually parse those phrases.

Following the definition of endocentricity, a phrase containing a head and an optional element

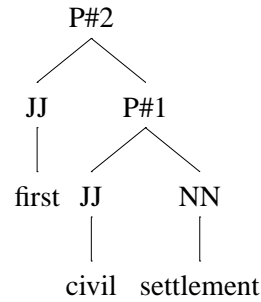


Figure 3: Structure without knowledge about endocentricity

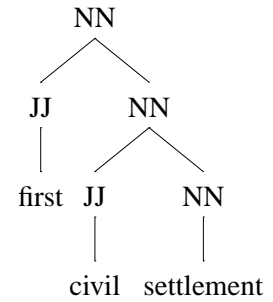


Figure 4: Structure using knowledge about endocentricity

should be equally distributed — in respect to its context — as the head. Consequentially, a phrase is considered to be endocentric, if it contains an element showing high context similarity (see Equ. 9).

$$\begin{aligned} \text{endocentric}(P) &\Leftrightarrow \\ &\exists i : \text{sim}(\text{context}(P), \text{context}(p_i)) \geq \vartheta \end{aligned} \quad (9)$$

The global context $\text{context}(P)$ of a phrase or POS tag P is the sum of all local contexts of P within the training corpus. We use the two left and right neighbours including the aforementioned markers for the beginning and the end of a sentence if necessary. We apply the Cosine Measure to calculate the similarity between the two contexts and in case of passing a defined threshold ϑ , the phrase is considered to be endocentric. See Table 2 for some examples ($\vartheta = 0.9$).

NNS	←	JJ NNS
NN	←	JJ NN
NNP	←	NNP CC NNP
NN	←	NN CC NN
VBZ	←	RB VBZ

Table 2: Examples of endocentric constructions

2.3 Discontiguous dependencies

Additionally to endocentric constructions containing a head and a modifier, some parts-of-speech like articles and possessive pronouns do not occur without a noun or noun phrase. While those parts-of-speech are grouped together as determiners (DT) in the Penn Tree Tagset, for other tagsets and languages they might be distributed among multiple classes (as in the German Stuttgart–Tübingen

Tagset among *ART*, *PPOSAT*, *PIAT* . . .). To detect such strong dependencies, we propose a simple test measuring the relative score of observing two words *A* and *B* together within a maximum range *n*.

$$dep_n(A, B) = \frac{\sum_{d=0}^n freq(A, B, d)}{\min(freq(A), freq(B))} \quad (10)$$

Equ. 10 formally describes the relative score where $freq(A, B, d)$ denotes the frequency of *A* and *B* occurring together with exactly *d* other tokens between them. If $dep_n(A, B)$ passes a threshold ϑ (0.9 for our experiments), then the dependency between *A* and *B* is allowed to occur discontinuously. Including these dependencies facilitates the parsing of rare and insignificant phrases like adjectival phrases.

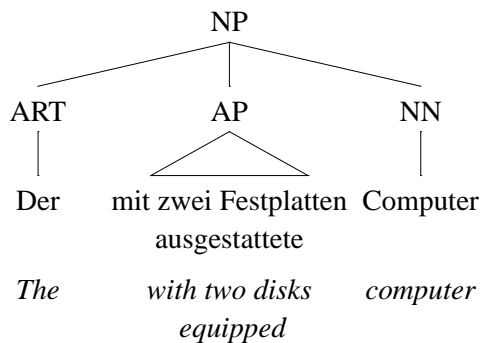


Figure 5: Adjectival Phrase

In the example given in Figure 5, the discontinuous dependency between articles (ART) and normal nouns (NN) can be applied to two possible word pairs. On the one hand, there is *Der . . . Festplatten* (*The . . . disks*), the other possibility is *Der . . . Computer* (*The . . . computer*). We choose the pair achieving the highest neighbourhood co-occurrence significance. Regarding our example, it is quite obvious that *Computer* is the noun to choose as *Der* and *Computer* show grammatical agreement while this is not the case for *Festplatten*. Consequently, the significance of *Der Computer* is much higher than the one of *Der Festplatten*. Although articles and other parts-of-speech are not unambiguous regarding gender, number and case for all languages, this approach can resolve some of those cases for certain languages.

2.4 Phrase Clustering

One objection to unsupervised parsing is the fact that phrases belonging to the same phrase type are

not labeled the same way. And of course, without any prior knowledge, induced phrases will never be labeled *NP*, *PP* or like any other known phrase type. This complicates the application of any further algorithms relying on that knowledge. Nevertheless, it is possible to cluster syntactic identical phrases into one class.

As in section 2.2, similarity between two global contexts is calculated. If the similarity of phrase *P* (the one being tested) and *Q* (see most similar one, see Equ. 11) exceeds a threshold ϑ , then phrase *P* is considered to have the same phrase type as *Q* (see Equ. 12). In this case, *P* will be labeled by the label of *Q* and thus, is treated like *Q*.

$$Q = \arg \max_{q \in phrases} sim(context(P), context(q)) \quad (11)$$

$$Type(P) = Type(Q) \Leftrightarrow sim(P, Q) \geq \vartheta \quad (12)$$

As it can be seen in Table 3 ($\vartheta = 0.9$), clustering finds syntactic similar phrases and facilitates iterative learning as rules can be learned for each phrase type and not for each composition.

P#1	←	DT JJ NN
P#1	←	DT NN
P#1	←	PRP\$ NNS
P#2	←	IN P#1
P#2	←	IN NN
P#2	←	IN NNS

Table 3: Results of phrase clustering

2.5 Iterative learning

Learning rules is realized as an iterative process. A flow chart of the proposed process is given in Figure 6.

First, an empty parser model is initialized. At the beginning of an iteration all rules are applied to transform the corpus. Resulting structures form the data which is used for the next iteration. The sentence in Figure 7 will be transformed by already induced rules.

After application of rule $NN \leftarrow JJ NN$, the optional element *JJ* is removed (see Fig. 8).

The next rule ($P\#1 \leftarrow DT NN$) reduces the complexity of the sentence and from now on, further rules will be created on those parts-of-speech and phrases (see Fig. 9).

Learning will be aborted after one of the following three conditions becomes true:

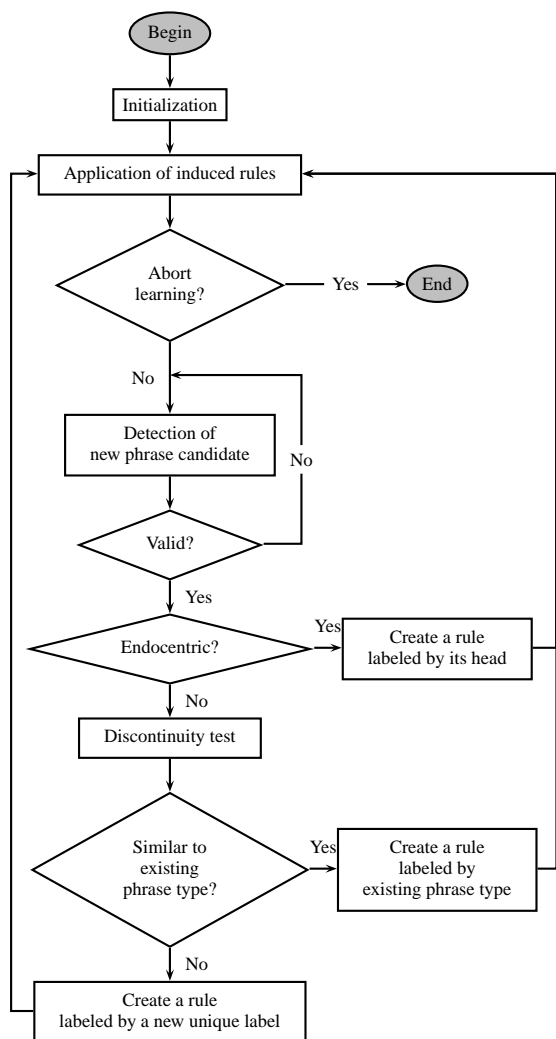


Figure 6: Flow chart of the proposed learning process

1. The algorithm reaches the maximum number of rules.
2. The last phrase candidate is not considered to be significant enough. A threshold in relation to the highest significance can be set up.
3. All sentences contained in the training corpus are reduced to one phrase.

Afterwards, the most significant n-gram passing the validity test will be regarded as a phrase. In the following steps, the label of the new phrase will be determined. Either it is labeled by its head (in case of an endocentric construction) or by a syntactic identical phrase type that has been learned before. If neither is the case, it gets a new unique label. Afterwards, the next iteration is triggered.

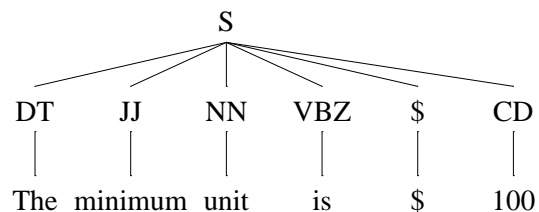


Figure 7: Example

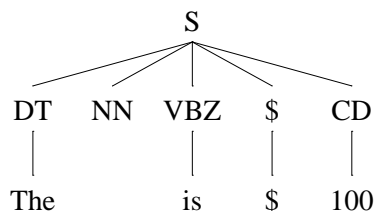


Figure 8: Example after application of rule $NN \leftarrow JJ NN$

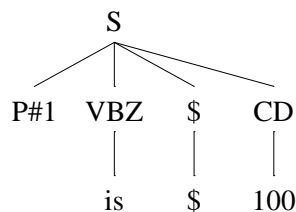


Figure 9: Example after additional application of rule $P\#1 \leftarrow DT NN$

3 Evaluation

To evaluate unSUParse+ against unSUParse and other unsupervised parsing algorithms, we apply the same experimental setup as in (Klein, 2005), (Bod, 2006) and (Hänig et al., 2008). For German punctuation and empty element tags are removed from the NEGRA corpus (see (Skut et al., 1998)). Afterwards, all sentences containing more than 10 elements are dismissed. The resulting corpus is referred to as NEGRA10 (2175 sentences). To take more complex sentences into account, we also prepared a corpus containing sentences to a maximum length of 40 elements (NEGRA40).

We present results for both — POS tags and word strings. As most unsupervised parsing models (except (Seginer, 2007)), we apply the hand-annotated data of the NEGRA corpus. Additionally, we used an unsupervised part-of-speech tagger (see (Biemann, 2006)) to tag the NEGRA corpus to be able to present a complete unsupervised parsing process relying on word strings only. We applied the model *de40M* which has been created

on a corpus containing 40 million sentences and contains 510 word classes.

To compare the performance of different parsing algorithms, we used the Unlabeled Brackets Measure as in (Klein and Manning, 2002) and (Klein and Manning, 2004). Additionally to unlabeled precision UP and unlabeled recall UR, the unlabeled f-score UF is defined as:

$$UF = \frac{2 \cdot UP \cdot UR}{UP + UR} \quad (13)$$

The baseline algorithm is based on neighbourhood co-occurrences. First, a parse tree is initialized and all tokens of a sentence are added as leaves. Afterwards, the two adjacent nodes containing the POS tags with the highest neighbourhood co-occurrence significance are merged consecutively until a binary tree has been created.

Results for NEGRA10 are given in Table 4. *unsuParse+* improves the performance of *unsuParse* in both categories: supervised and unsupervised annotated POS tags. While recall is improved significantly for hand-annotated data, just a slight improvement is achieved for word strings. Especially clustering of phrases leads to the increased recall as rules do not need to be learned for every possible compound composition of a given phrase type as they are already covered by the phrase type itself. Models based on *unsuParse* achieve the highest precision among all models. This is not very surprising as most of the other models (except Common Cover Links) generate binary parses achieving a higher recall. Nevertheless, *unsuParse+* yields comparable results and obtains the highest f-score for German data.

Parsing Model	UP	UR	UF
Baseline (POS tags)	35.5	66.0	46.2
CCM	48.1	85.5	61.6
DMV + CCM	49.6	89.7	63.9
U-DOP	51.2	90.5	65.4
UML-DOP	—	—	67.0
U-DOP*	—	—	63.8
<i>unsuParse</i> (POS tags)	76.9	53.9	63.4
<i>unsuParse+</i> (POS tags)	71.1	67.9	69.5
Baseline (words)	23.6	43.9	30.7
Common Cover Links	51.0	69.8	59.0
<i>unsuParse</i> (words)	61.2	59.1	60.2
<i>unsuParse+</i> (words)	63.1	60.4	61.7

Table 4: UP, UR and UF for NEGRA10

Performance drops for more complex sentences

(see Table 5). As for short sentences, the recall of our approach is in the same order as for the baseline. However, precision is increased by a factor of two in comparison to the baseline, which is also similar to short sentences.

Parsing Model	UP	UR	UF
Baseline (POS tags)	24.8	49.3	33.0
<i>unsuParse+</i> (POS tags)	55.3	51.4	53.3

Table 5: UP, UR and UF for NEGRA40

Table 6 shows the most frequently over- and under-proposed phrases for NEGRA10. Noun and prepositional phrases are often over-proposed due to a flat representation within the NEGRA corpus. The most frequently under-proposed phrase *NE NE* is learned and classified as endocentric construction ($NE \leftarrow NE NE$). Due to the removal of punctuation, proper nouns which naturally would be separated by e. g. commas will be represented by one flat phrase without deeper analysis of the inner structure. This includes some underpropositions which will not occur while parsing sentences containing punctuation.

Overproposed		Underproposed	
ART NN	369	NE NE	42
CARD NN	111	NN NE	35
ADV ADV	103	ART NN NE	27
ADJA NN	99	ADV ART NN	24
APPR ART NN	93	APPR PPER	23

Table 6: Most frequently over- and under-proposed constituents

4 Conclusions and further work

In this paper, we presented an improved model for co-occurrence based parsing. This model creates high accuracy parses employing a constituent detection algorithm yielding competitive results. Although no a priori knowledge about the language in question is taken into account, it is possible to detect heads, modifiers and different phrase types. Especially noun phrases and prepositional phrases are clustered into their respective classes. For further processing like relation extraction, precise results for the aforementioned phrase types are essential and provided by this algorithm in an unsupervised manner.

Our future work will include the investigation of unsupervised methods for dependency identifica-

tion between verbs and their arguments. Furthermore, the inclusion of further constituency tests like substitution and deletion could provide additional certainty for constituent candidates.

References

- David Adger. 2003. *Core Syntax: A Minimalist Approach*. Oxford University Press.
- Chris Biemann. 2006. Unsupervised part-of-speech tagging employing efficient graph clustering. In *Proceedings of the COLING/ACL-06 Student Research Workshop*, Sydney, Australia.
- Rens Bod. 2006. An all-subtrees approach to unsupervised parsing. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*.
- Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, Massachusetts.
- Ted Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.
- Christian Hänig, Stefan Bordag, and Uwe Quasthoff. 2008. Unsuparse: Unsupervised parsing with unsupervised part of speech tagging. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*.
- Richard S. Kayne. 1995. *The Antisymmetry of Syntax*. MIT Press.
- Dan Klein and Christopher D. Manning. 2002. A generative constituent-context model for improved grammar induction. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*.
- Dan Klein and Christopher D. Manning. 2004. Corpus-based induction of syntactic structure: models of dependency and constituency. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*.
- Dan Klein. 2005. *The Unsupervised Learning of Natural Language Structure*. Ph.D. thesis, Stanford University.
- Ewald Lang. 2002. Die Wortart "Konjunktion". In *Lexikologie. Lexicology. Ein Internationales Handbuch zur Natur und Struktur von Wörtern und Wortschätzen*, pages 634–641. de Gruyter.
- M. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.
- Carl Pollard and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University Of Chicago Press.
- Ivan Sag. 2002. Coordination and underspecification. In *Proceedings of the Ninth International Conference on Head-Driven Phrase Structure Grammar*.
- Yoav Seginer. 2007. Fast unsupervised incremental parsing. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*.
- Wojciech Skut, Thorsten Brants, Brigitte Krenn, and Hans Uszkoreit. 1998. A linguistically interpreted corpus of german newspaper text. In *ESSLLI-98 Workshop on Recent Advances in Corpus Annotation*.
- C. Thielen, A. Schiller, S. Teufel, and C. Stöckert. 1999. Guidelines für das Tagging deutscher Textkorpora mit STTS. Technical report, University of Stuttgart and University of Tübingen.