Unsupervised Induction of Labeled Parse Trees by Clustering with Syntactic Features

Roi Reichart
ICNC
Hebrew University of Jerusalem
roiri@cs.huji.ac.il

Ari Rappoport
Institute of computer science
Hebrew University of Jerusalem
arir@cs.huji.ac.il

Abstract

We present an algorithm for unsupervised induction of labeled parse trees. The algorithm has three stages: bracketing, initial labeling, and label clustering. Bracketing is done from raw text using an unsupervised incremental parser. Initial labeling is done using a merging model that aims at minimizing the grammar description length. Finally, labels are clustered to a desired number of labels using syntactic features extracted from the initially labeled trees. The algorithm obtains 59% labeled f-score on the WSJ10 corpus, as compared to 35% in previous work, and substantial error reduction over a random baseline. We report results for English, German and Chinese corpora, using two label mapping methods and two label set sizes.

1 Introduction

Unsupervised learning of grammar from text (‘grammar induction’) is of great theoretical and practical importance. It can shed light on language acquisition by humans and on the general structure of language, and it can potentially assist NLP applications that utilize parser output. The problem has attracted researchers for decades, and interest has greatly increased recently, in part due to the availability of huge corpora, computation power, and new learning algorithms (see Section 2).

A fundamental issue in this research direction is the representation of the resulting induced grammar. Most recent work (e.g., (Klein and Manning, 2004; Dennis, 2005; Bod, 2006a; Smith and Eisner, 2006; Seginer, 2007)) annotates text sentences using a hierarchical bracketing (constituents) or a dependency structure, and thus represents the induced grammar through its behavior in a parsing task. Solan et al. (2005) uses a graph representation, while (Nakamura, 2006) simply uses a grammar formalism such as PCFG. When the bracketing approach is taken, some algorithms label the resulting constituents, while most do not.

Each of these approaches can be justified or criticized; a detailed discussion of this issue is beyond the scope of this paper. The algorithm presented here belongs to the first group, annotating given sentences with labeled bracketing structures. The main theoretical justification for this approach is that many linguistic and psycho-linguistic theories posit some kind of a hierarchical labeled constituent (or constructional) structure, arguing that it has a measurable psychological (cognitive) reality (e.g., (Goldberg, 2006)). The main practical arguments in favor of this approach are that it enables a detailed and large-scale evaluation using annotated corpora, as is done in this paper, and that the output format is suitable for many applications.

When an algorithm generates labeled structures, the number of labels is an important issue. From a theoretical point of view, the algorithm should also discover the appropriate number of labels. However, for evaluation and application purposes it is useful to base the number of labels on a specific *target grammar*. In previous work, the number was set to be equal to that in the target grammar. This is a reasonable approach that we experiment with in this paper. In addition, to reduce the possible arbitrariness in this approach, we also experiment with the number of *prominent labels* in the target

Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 721–728
Manchester, August 2008

grammar, determined according to their coverage of corpus constituents.

Another issue relates to the nature of the input. In most cases (e.g., in the Klein, Smith, Dennis and Bod papers above), the input consists of part-of-speech (POS) sequences, derived from text corpora by manual or automatic POS tagging. In some cases (e.g., in the Seginer and Solan papers above) it can consist of plain text. Again, each approach has its pros and cons. The algorithm we present here requires POS tags for its labeling stages. Parts-of-speech are widely considered to have a psychological reality (at least in English, including when they are viewed as low-level constructions as in (Croft, 2001)), so this kind of input is reasonable for theoretical research. Moreover, as POS induction is of medium quality (Clark, 2003), using a manually POS tagged corpus enables us to measure the performance of other induction stages in a controlled manner. Since supervised POS tagging is of very high quality and very efficient computationally (Brants, 2000), this requirement does not seriously limit the practical applicability of a grammar induction algorithm.

Our labeled bracketings induction algorithm consists of three stages. We first induce unlabeled bracketing trees using the algorithm given in (Seginer, 2007)[1]. We then induce initial labels using a *Bayesian Model Merging (BMM)* labeling algorithm (Borensztajn and Zuidema, 2007), which aims at minimizing the description length of the input data and the induced grammar. Finally, the initial labels are clustered to a desired number of labels using syntactic features extracted from the initially labeled trees. Previous work on labeled brackets induction (Section 2) did not differentiate the unlabeled structure induction phase from the labeling phase, applying a single phase approach.

To evaluate labeled bracketings, we need a mapping between the label symbols of the induced and target grammars. Previous work used a 'greedy', many to one, mapping. We used both the greedy mapping and a label-to-label (LL) mapping, since greedy mapping is highly forgiving to structural problems in the induced labeling. We report results for two cases: one in which the number of labels in the induced and target grammars is the same, and one in which the former is the number of prominent labels in the target grammar. We discuss how this number can be defined and determined. We experimented with English (WSJ10, Brown10), German (NEGRA10) and Chinese (CTB10) corpora.

When comparing to previous work that used manually annotated corpora in its evaluation (Haghighi and Klein, 2006)[2], we obtained 59.5% labeled f-score on the WSJ10 setup vs. their 35.3% (Section 5). We also show substantial improvement over a random baseline, and that the clustering stage of our algorithm improves the results of the second merging stage.

Section 2 discusses previous work. In Section 3 we detail our algorithm. The experimental setup and results are presented in Sections 4 and 5.

2 Previous Work

Unsupervised parsing has attracted researchers for decades (see (Clark, 2001; Klein, 2005) for recent reviews). Many types of input, syntax formalisms, search procedures, and success criteria were used. Among the theoretical and practical motivations to this problem are the study of human language acquisition (in particular, an empirical study of the poverty of stimulus hypothesis), preprocessing for constructing large treebanks (Van Zaanen, 2001), and improving language models (Chen, 1995).

In recent years efforts have been made to evaluate the algorithms on manually annotated corpora such as the WSJ PennTreebank. Recently, works along this line have for the first time outperformed the right branching heuristic baseline for English. These include the constituent–context model (CCM) (Klein and Manning, 2002), its extension using a dependency model (Klein and Manning, 2004), (U)DOP based models (Bod, 2006a; Bod, 2006b; Bod, 2007), an exemplar–based approach (Dennis, 2005), guiding EM using contrastive estimation (Smith and Eisner, 2006), and the incremental parser of (Seginer, 2007). All of these use as input POS tag sequences, except of Seginer's algorithm, which uses plain text. All of these papers induce unlabeled bracketing or dependencies.

There are other algorithmic approaches to the problem (e.g., (Adriaans, 1992; Daelemans, 1995; Van Zaanen, 2001)). None of these had evaluated labeled bracketing on annotated corpora.

In this paper we focus on the induction of *labeled* bracketing. Bayesian Model Merging

[1] The algorithm uses raw (not POS tagged) sentences.

[2] Using, as they did, a greedy mapping with an equal number of labels in the induced and target grammars.

(BMM) (Stolcke, 1994; Stolcke and Omohundro, 1994) is a framework for inducing PCFG containing both a bracketing and a labeling. The characteristics of this framework (separating prior probability, data likelihood and heuristic search procedures) can also be found in the grammar induction models of (Wolf, 1982; Langley and Stromsten, 2000; Petasis et al., 2004; Solan et al., 2005). The BMM model used here (Borensztajn and Zuidema, 2007) combines features of (Petasis et al., 2004) and Stolcke's algorithm, applying the minimum description length (MDL) principle. We use it here only for initial labeling of existing bracketings. The MDL principle was also used in (Grunwald, 1994; de Marcken, 1995; Clark, 2001).

There are only two previous papers we are aware of that induce labeled bracketing and evaluate on corpora annotated with a similar representation (Haghighi and Klein, 2006; Borensztajn and Zuidema, 2007). We utilize and extend the latter's labeling algorithm. However, the evaluation done by the latter dealt only with labeling, using gold-standard (manually annotated) bracketings. Thus, we can directly compare our results only to (Haghighi and Klein, 2006), where two models (*PCFG* × *NONE* and *PCFG* × *CCM*) are fully unsupervised. These models use the inside-outside and EM algorithms to induce bracketing and labeling simultaneously, as opposed to our three step method[3].

3 Algorithm

Our model consists of three stages: bracketing, initial labeling, and label clustering.

3.1 Induction of Unlabeled Bracketing

In this step, we apply the algorithm of (Seginer, 2007) to induce bracketing from plain text[4]. We have chosen that algorithm because it is very fast (both learning and parsing) and its code is publicly available. We could have chosen any of the algorithms mentioned above producing a similar output format.

3.2 Initial Constituent Labeling

Our label clustering stage uses syntactic features. To obtain these, we need an initial labeling on the bracketings computed in the previous stage. To do that we modify the Bayesian Model Merging (BMM) algorithm of (Borensztajn and Zuidema, 2007), which induces context-free grammars (bracketing and labeling) from POS tags, combining features of the models of (Stolcke and Omohundro, 1994) and (Petasis et al., 2004).

The BMM algorithm (Borensztajn and Zuidema, 2007) uses an iterative heuristic greedy search for an optimal PCFG according to the Bayesian criterion of maximum posterior probability. Two operators define possible transitions between grammars: MERGE creates generalizations by replacing two existing non-terminals X_1 and X_2 that occur in the same contexts by a single new non-terminal Y; CHUNK concatenates repeating patterns by taking a sequence of two non-terminals X_1 and X_2 and creating a new non-terminal Y that expands to X_1X_2.

We have used the algorithm to deal only with labeling. It reads the initial rules of the grammar from all productions implicit in the bracketed corpus induced in the previous step. Every constituent (except of the start symbol) is given a unique label. Since only labeling is required, only MERGE operations are performed.

The objective function the algorithm tries to optimize at each step is the posterior probability calculated according to Bayes' Law:

$$M_{MAP} = argmax_M \; P(M|X) = argmax_M \; P(X|M) \cdot P(M) \tag{1}$$

where $P(X|M)$ is the likelihood of the data X given the grammar M and $P(M)$ is the prior probability of the grammar. This is equivalent to minimizing the function

$$-log(P(X|M)) - logP(m) := DDL + GDL := DL. \tag{2}$$

Using a Minimal Description Length (MDL) principle, BMM interprets this function as total description length (DL): The Grammar Description Length $GDL = -logP(M)$ is the space needed to encode the model, and the Data Description Length $DDL = -logP(X|M)$ is the space required to describe the data given the model. The rationale for MDL is to prefer smaller grammars that describe the data well. DDL and GDL are computed as in (Stolcke, 1994; Stolcke and Omohundro, 1994). In order to reduce the number of grammars considered at each step, which naively is quadratic in the number of non-terminals, a method based on (Petasis et al., 2004) for efficiently predicting DL gain is applied. The process

[3] Their other models, which were the core of their paper, are semi-supervised.

[4] http://www.seggu.net/ccl

is iterated until no additional merge operation improves the objective function. Full details are given in (Borensztajn and Zuidema, 2007).

3.3 Label Clustering

Label set size. BMM produces quite a large number of labels (4944 for WSJ10[5]). In the third step of our algorithm we reduce that number. We first discuss the issue of the number of labels in induced grammars, which is an important issue.

In many situations, it is reasonable to use a number T identical to the number of labels in a given target grammar, for example when that grammar is used for applications or evaluation. This is the approach in (Haghighi and Klein, 2006) for their unsupervised models[6], and we use it in part of our evaluation. However, it is also reasonable to argue that the granularity of syntactic categories (labels) in the gold standard annotation of the corpora we experiment with is somewhat arbitrary. For example, in the WSJ Penn Treebank noun phrases are annotated with the symbol NP, but there is no distinction between subject and object NPs. Incorporating such a distinction into the WSJ10 grammar would result in a 27 labels grammar instead of 26.

To examine this issue, consider Figure 1, which shows the amount of constituent coverage obtained by a certain number of labels in the four corpora we use (see Section 4). In all of them, about 95% of the constituents are covered by 23% – 37% of the labels, and the curve rises very sharply until that 95% value. Motivated by this observation, given a corpus annotated using a certain hierarchical labeled grammar, we refer to the set of P labels that cover at least 95% of the constituents in the corpus as the grammar's *prominent* labels.

The prominent labels are not only the most frequent in the corpus; each of them substantially contributes to constituent labeling, while the saliency of other labels is much smaller. It is thus reasonable to assume that by addressing only prominent labels, we address a level of granularity that is uniform and basic (to the annotation scheme used). As a result, by asking the induced grammar to produce P labels, we reduce arbitrariness and enable our testing to focus on our success in identifying the basic phenomena in the target grammar.

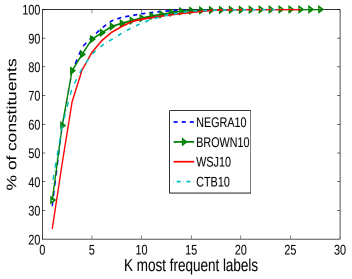

Figure 1: For each k, the fraction of constituents labeled with the k most frequent labels, for WSJ10 (solid), Brown10 (triangles), NEGRA10 (dashed) and CTB10 (dotted). In all corpora, more than 95% of the constituents are labeled using less than 10 *prominent* labels.

As a result, we generated two grammars for each corpus we experimented with, one having T labels and the other having P labels.

Clustering. we stop BMM when no improvement to its objective function is possible, and cluster the labels to conform to the size constraint. [7]

Denote the number of labels in the induced grammar with M, the set of D most frequent induced labels with A, and the set consisting of the other induced labels with B ($|B| = M - D$). If $M \not> D$, there is nothing to do since the constraint holds. Otherwise, we map each label in B to the label in A that exhibits the most similar syntactic behavior, as follows. We construct a feature vector representation of each of the labels, using $3M + |K|$ features, where K is the set of POS tags in the corpus. The first M features correspond to parent-child relationships between each of the induced labels and the represented label. The i-th feature ($i \in [1, M]$) is the number of times the i-th label is the parent of the represented label. Similarly, the next M features correspond to child-parent relationships, the next M features correspond to sibling relationships and the last $|K|$ features correspond to the number of times each POS tag is the leftmost POS tag in a constituent labeled by the represented label. Note that in order to compute the values of the first $3M$ features, we needed an initial labeling on the induced bracketings; this is the main reason for using the BMM stage.

For each label $b_i \in B$, we compute the cosine

[5] For completeness, in Section 5 we provide results for this grammar using greedy mapping evaluation. LL mapping evaluation cannot be performed when the numbers of induced and target labels differ.

[6] Personal communication with the authors.

[7] It is possible to force BMM to iterate until a desired number of induced labels (T or P) is achieved. However, the induced grammars are of very low quality (see Section 5).

metric between its vector b_i^v and that of every $a_j \in A$, mapping b_i to the label a_j with which it obtains the highest score:

$$Map(b_i) = argmax_j \frac{b_i^v \cdot a_j^v}{|b_i^v||a_j^v|} \quad (3)$$

The cosine metric grows when the same coordinates (features) in both vectors have higher values. As a result, vectors with high values of the same features (corresponding to similar syntactic behavior) get high scores.

4 Experimental Setup

We evaluated our algorithm on English, German and Chinese corpora: the WSJ Penn Treebank, containing economic English newspaper articles, the Brown corpus, containing various English genres, the Negra corpus (Brants, 1997) of German newspaper text, and version 5.0 of the Chinese Penn Treebank (Xue et al., 2002). In each corpus, we used the sentences of length at most 10[8], numbering 7422 (WSJ10), 9117 (Brown10), 7542 (NEGRA10) and 4626 (CTB10).

For each corpus the following T and P values were used: WSJ10: $26, 8$; Brown10: $28, 7$; NEGRA10: $22, 6$; CTB10: $24, 9$. Each number produces a different grammar.

For labeled f-score evaluation, the induced labels should be mapped to the target labels[9]. We evaluated with two different mapping schemes. For each pair (X_i, Y_j) of induced and target labels, let C_{X_i,Y_j} be the number of times they label a constituent having the same span in the same sentence. Following (Haghighi and Klein, 2006) we applied a greedy (many to one) mapping where the mapping is given by $Map(X_i) = argmax_{Y_j}\ C_{X_i,Y_j}$. This greedy mapping tends to map many induced labels to the same target label, and is therefore highly forgiving of large mismatches between the structures of the induced and target grammars. Hence, we also applied a label-to-label (LL) mapping, computed by reducing this problem to optimal assignment in a weighted complete bipartite graph, formally defined as follows. Given a weighted complete bipartite graph $G = (X \cup Y; X \times Y)$ where edge (X_i, Y_j) has weight w_{ij}, find a (one-to-one) matching M from X to Y having a maximal weight. In our case, X is the set of model symbols, Y is the set of T or P most frequent target symbols (depending on the desired label set size used), and $w_{ij} := C_{X_i,Y_j}$, computed as in greedy mapping (the number of times x_i and y_j share a constituent). To make the graph complete, we add zero weight edges between induced and target labels that do not share any constituent. The Kuhn-Munkres algorithm (Kuhn, 1955; Munkres, 1957) solves this problem, and we used it to perform the LL mapping (see also (Luo, 2005)).

We assessed the overall quality of our algorithm, the quality of its labeling stage and the quality of the syntactic clustering (SC) stage. For the overall quality of the induced grammar (both bracketing and labeling) we compare our results with (Haghighi and Klein, 2006), using their setup[10]. That setup was used for all numbers reported in this paper. Note that a random baseline would yield very poor results, so there is nothing to be gained from comparing to it.

We assessed the quality of the labeling (MDL and SC) stages alone, using only the correct bracketings produced by the first stage of the algorithm. We compare to a *random baseline* on these correct constituents that randomly selects (using a uniform distribution) a label for each constituent among the set of labels allowed to the algorithm.

To asses the quality of the third stage (SC) we compare the f-score performance of our three stages labeled trees induction algorithm (bracketing, MDL, SC) to an algorithm consisting of the first two stages only (bracketing and MDL) and the accuracy of the two stages labeling algorithm (MDL, SC) to an algorithm where the syntactic clustering stage is replaced by a simpler method (MDL, random clustering).

5 Results

We start with comparing our algorithm with (Haghighi and Klein, 2006), the only previous work that produces labeled bracketing and was tested on large manually annotated corpora. Their relevant models are PCFG × NONE and PCFG × CCM[11].

[8] Excluding punctuation and null elements, according to the scheme of (Klein, 2005).

[9] There are many possible methods for evaluating clustering quality (Rosenberg and Hirschberg, 2007). For our task, overall f-score is a very natural one. We will address other methods in future papers.

[10] Brackets covering a single word are not counted, multiple labels and the sentence level constituent are counted. Two sentence level constituents are usually used: one for the root symbol at the top (which was not counted), and one real symbol (in WSJ10 it is usually, but not always, S), which was counted. We had verified the setup with the authors.

[11] They focused on a different, semi-supervised, setting.

	This Paper	*PCFG* × *CCM*	*PCFG* × *NONE*
WSJ10	**59.5**	35.3	26.3

Table 1: F-scores of our algorithm and of the unsupervised models in (Haghighi and Klein, 2006) on WSJ10 (they did not test these models on the other corpora we experimented with).

The number of labels in their induced grammar equals the number of labels in the target grammar (26 for WSJ10), and they had used a greedy mapping. Table 1 shows that our algorithm achieves a superior f-score of 59.5% over their 35.3%. Haghighi and Klein (2006) did not experiment with the NEGRA10 and Brown10 corpora, and had used version 3.0 of CTB10 while we have used the substantially different version 5.0, so we can only compare our results on WSJ10.

Table 2 shows the labeled recall, precision and f-score of our algorithm on the various corpora and mappings we use. On Brown10, NEGRA10 and CTB10 (version 5.0) these are the first reported results for this task. For reference, the table also shows the unlabeled f-score results of Seginer's bracketing algorithm (our first stage)[12].

We can see that greedy mapping is indeed more forgiving than LL mapping, for both T labels and P labels. WSJ results are generally higher than for the other corpora, probably because WSJ bracketing results are higher than for the other corpora.

Comparing the left and right columns in each of the table sections reveals that for greedy mapping, mapping to a higher number of labels results in higher scores than mapping to a lower number. LL mapping behaves in exactly the opposite way. The explanation for this is that when we force the mapping to cover all of the target labels (as done by LL mapping for T labels), we move probability mass from the correct, heavy labels to smaller ones, thereby magnifying errors.

Table 4 addresses the quality of the whole labeling stage (MDL and SC) and of the SC stage. We report the quality of our labels (top line for each corpus in the table) the random baseline labels (third line) and the labels of an algorithm where MDL is performed and the syntactic clustering is replaced by a random clustering (RC) algorithm that, given a label L that is not one of the T or P most frequent labels, randomly selects one of the most frequent labels and adds L to its cluster (second line).[13] All three labeling algorithms used Seginer's bracketing and results are reported only for labels of correctly bracketed constituents. Reported are the algorithm and baselines accuracy (percentage of correctly labeled constituents after the mapping has been performed) and the error reduction of the algorithm over the baselines (bottom line). (MDL,SC) substantially outperforms both the random baseline, demonstrating the power of the whole labeling stage, and the (MDL,RC) algorithm, demonstrating the power of the SC stage.

	Greedy		LL	
	T	P	T	P
WSJ10				
MDL,SC	80	67	47	59
MDL,RC	67	61	37	42
Rand. Base.	30	30	5	14
Error Reduction	39%,71%	15%,53%	16%, 44%	29%, 52%
Brown10				
MDL,SC	73	61	48	60
MDL,RC	68	59	46	51
Rand. Base.	27	27	4	14
Error Reduction	16%,63%	5%, 47%	4%, 46%	18%, 53%
NEGRA10				
MDL,SC	79	72	65	72
MDL,RC	73	69	54	58
Rand. Base.	39	39	5	17
Error Reduction	22%,66%	10%,34%	24%,63%	33%,66%
CTB10				
MDL,SC	70	67	44	55
MDL,RC	36	32	40	45
Rand. Base.	29	29	5	12
Error Reduction	53%,58%	51%, 54%	7%,41%	18%,49%

Table 4: Pure labeling results (taking into account only the correct bracketings produced at stage 1), compared to the random and (MDL,RC) baselines. The left number in the Error Reduction lines slots compares (MDL,SC) to (MDL,RC) and the right number compares (MDL,SC) to random labeling. (MDL,SC) algorithm is substantially superior.

We compared our grammars to the grammars induced by the first two stages (bracketing and then MDL that stops when no DL improvement is possible) alone. Since the number of labels in these grammars is much larger than in the target grammar, only the evaluation with the greedy, many to one, mapping is performed. Using greedy mapping, the F-score of these grammars constitutes an upper bound on the F-score after the subsequent SC stage. For WSJ10 (4944 labels), NEGRA10 (5557 labels), CTB10 (2298 labels) and Brown10 (3314 labels) F-score values are 64.6, 49.9, 38.7 and 52.5 compared to F-score values of 59.5(50.2), 45.6(42), 36.4(34.7) and 49.4(41.3) after mapping all induced labels to the T (P) most frequent labels with SC (Table 2, 'greedy' section). The frac-

[12] The numbers slightly differ from those in Seginer's paper, since we use the (Haghighi and Klein, 2006) setup.

[13] Our algorithm's numbers can be deduced from Table 2. Results for all random baselines are averaged over 10 runs.

	Greedy Mapping						LL Mapping						Seginer
	T labels			P labels			T labels			P labels			(unlabeled)
Corpus	R	P	F	R	P	F	R	P	F	R	P	F	F
WSJ10	58	61	59.5	48.9	51.5	50.2	34.2	36.1	35.2	42.7	44.9	43.8	74.6
NEGRA10	54.2	39.3	45.6	50	36.2	42	44.7	32.4	37.6	49.5	35.9	41.7	58.1
CTB10	35.1	37.8	36.4	33.4	36	34.7	21.9	23.6	22.7	27.4	29.5	28.4	51.8
Brown10	47.6	51.3	49.4	39.9	43	41.3	31.3	33.7	32.4	38.9	41.9	40.3	67.8

Table 2: Labeled recall, precision and f-score of our algorithm, mapping model labels into target labels greedily (left) and using LL mapping (right). The number of induced labels was set to be the total number T of target labels or the number P of prominent labels in the target grammar (WSJ10: 26, 8; Brown10: 28, 7; NEGRA10: 22, 6; CTB10: 24, 9). Also shown are Seginer's unlabeled bracketing results (rightmost column), which constitute an upper bound on the quality of subsequent labeling steps.

	WSJ10						Brown10					
Label	T labels			P labels			T labels			P labels		
	R	P	F	R	P	F	R	P	F	R	P	F
S	77.1	77.6	77.3	75.4	67.9	71.5	72.3	60.9	66.1	69.3	63.2	66.1
NP	8.5	79.5	15.4	19.8	61.6	30	10.7	79.3	18.9	15.6	78	26
VP	20.4	67.6	31.3	64.2	36.7	46.7	9.8	72.5	17.3	14.1	59	22.8
PP	40.8	63.5	49.7	8	8.9	8.4	17.4	59.2	26.9	75.5	14.4	24.2

Table 3: Recall, Precision and F-score for constituents labeled with the 4 most frequent labels in the WSJ10 and Brown10 test sets. LL mapping is used for evaluation.

tion of constituents covered by the T (P) most frequent labels before mapping with SC is 0.42(0.29), 0.33(0.23), 0.58(0.45) and 0.66(0.42), emphasizing the effect of SC on the final result.

MDL finds the best merge at each iteration. Instead of stopping it when no DL gains are possible, we can keep merging after the deltas become worse than the total DL, stopping only when the desired number of labels (T or P) is achieved. We tried this version of a (bracketing and MDL) algorithm and obtained grammars of very low quality. This further demonstrates the importance of the SC stage.

Table 3 shows results for the four most frequent labels of WSJ10 and Brown10 .

6 Conclusion

Unsupervised grammar induction is a central research problem, possessing both theoretical and practical significance. There is great value in producing an output format consistent with and evaluated against formats used in large human annotated corpora. Most previous work of that kind produces unlabeled bracketing or dependencies. In this paper we presented an algorithm that induces labeled bracketing. The labeling stages of the algorithm use the MDL principle to induce an initial, relatively large, set of labels, which are then clustered using syntactic features. We discussed the issue of the desired number of labels, and introduced the concept of prominent labels, which allows us coverage of the basic and most salient level of a target grammar. Labels are clearly an important aspect of grammar induction. Future work will explore their significance for applications.

Evaluating induced labels is a complex issue. We applied greedy mapping as in previous work, and showed that our algorithm significantly outperforms it. In addition, we introduced LL mapping, which overcomes some of the shortcomings of greedy mapping. There are several other possible methods for evaluating labeled induced grammars, and we plan to explore them in future work. We evaluated on large human annotated corpora of different English domains and three languages, and showed that our labeling stages, and specifically the SC stage, outperform several baselines for all corpora and mapping methods.

Acknowledgments. We thank Gideon Borensztajn and Yoav Seginer for their help.

References

Pieter Adriaans, 1992. *Learning Language from a Categorical Perspective.* Ph.D. thesis, University of Amsterdam.

Rens Bod, 2006a. An All-Subtrees Approach to Unsupervised Parsing. *Proc. of the 44th Meeting of the ACL.*

Rens Bod, 2006b. Unsupervised Parsing with U-DOP. *Proc. of CoNLL X.*

Rens Bod, 2007. Is the End of Supervised Parsing in Sight? *Proc. of the 45th Meeting of the ACL.*

Gideon Borensztajn and Willem Zuidema, 2007. Bayesian Model Merging for Unsupervised Constituent Labeling and Grammar Induction. Technical Report, ILLC . http: //staff.science.uva.nl/∼gideon/

Thorsten Brants, 1997. The NEGRA Export Format. *CLAUS Report, Saarland University*.

Thorsten Brants, 2000. TnT: A Statistical Part-Of-Speech Tagger. *Proc. of the 6th Applied Natural Language Processing Conference*.

Stanley F. Chen, 1995. Bayesian grammar induction for language modeling. *Proc. of the 33th Meeting of the ACL*.

Alexander Clark, 2001. *Unsupervised Language Acquisition: Theory and Practice*. Ph.D. thesis, University of Sussex.

Alexander Clark, 2003. Combining Distributional and Morphological Information for Part of Speech Induction. *Proc. of the 10th Meeting of the European Chapter of the ACL.*

Willliam A. Croft, 2001. *Radical Construction Grammar*. Cambridge University Press.

Carl G. de Marcken, 1995. *Unsupervised Language Acquisition*. Ph.D. thesis, MIT.

Walter Daelemans, 1995. Memory-based lexical acquisition and processing. *Lecture Notes In Artificial Intelligence*, 898:85–98.

Simon Dennis, 2005. An exemplar-based approach to unsupervised parsing. *Proceedings of the 27th Conference of the Cognitive Science Society*.

Adele E. Goldberg, 2006. *Constructions at Work*. Oxford University Press.

Peter Grunwald, 1994. A minimum description length approach to grammar inference. *Lecture Notes In Artificial Intelligence*, 1004 : 203-216.

Aria Haghighi and Dan Klein, 2006. Prototype-driven grammar induction. *Proc. of the 44th Meeting of the ACL*.

Jin–Dong Kim, Tomoko Ohta, Yuka Teteisi and Jun'ichi Tsujii, 2003. GENIA corpus – a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19:i180–i182, Oxford University Press, 2003.

Dan Klein and Christopher Manning, 2002. A generative constituent-context model for improved grammar induction. *Proc. of the 40th Meeting of the ACL.*

Dan Klein and Christopher Manning, 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. *Proc. of the 42nd Meeting of the ACL.*

Dan Klein, 2005. *The unsupervised learning of natural language structure.* Ph.D. thesis, Stanford University.

Harold W. Kuhn, 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83-97.

Pat Langley and Sean Stromsten, 2000. Learning context-free grammars with a simplicity bias. *Proc. of the 11th European Conference on Machine Learning*.

Xiaoqiang Luo, 2005. On coreference resolution performance metrics. *Proc. of the 2005 Conference on Empirical Methods in Natural Language Processing*.

James Munkres, 1957. Algorithms for the Assignment and Transportation Problems. *Journal of the SIAM*, 5(1):32–38.

Katsuhiko Nakamura, 2006. Incremental learning of context free grammars by bridging rule generation and search for semi-optimum rule sets. *Proc. of the 8th ICGI.*

Georgios Petasis, Georgios Paliouras and Vangelis Karkaletsis, 2004. E-grids: Computationally efficient grammatical inference from positive examples. *Grammars*, 7:69–110.

Andrew Rosenberg and Julia Hirschberg, 2007. Entropy-based external cluster evaluation measure. *Proc. of the 2007 Conference on Empirical Methods in Natural Language Processing*.

Yoav Seginer, 2007. Fast Unsupervised Incremental Parsing. *Proc. of the 45th Meeting of the ACL.*

Noah A. Smith and Jason Eisner, 2006. Annealing Structural Bias in Multilingual Weighted Grammar Induction . *Proc. of the 44th Meeting of the ACL.*

Zach Solan, David Horn, Eytan Ruppin, and Shimon Edelman, 2005. Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences*, 102 : 11629–11634.

Andreas Stolcke. 1994. *Bayesian Learning of Probabilistic Language Models.* Ph.D. thesis, University of of California at Berkeley.

Andreas Stolcke and Stephen M. Omohundro, 1994. Inducing probabilistic grammars by Bayesian model merging . *Proc. of the 2nd ICGI.*

Menno van Zaanen, 2001. *Bootstrapping Structure into Language: Alignment-Based Learning.* Ph.D. thesis, University of Leeds.

J. Gerard Wolff, 1982. Language acquisition, data compression and generalization. *Language and Communication*, 2(1): 57–89.

Nianwen Xue, Fu-Dong Chiou and Martha Palmer, 2002. Building a large–scale annotated Chinese corpus. *Proc. of the 40th Meeting of the ACL.*