Generalizing a Strongly Lexicalized Parser using Unlabeled Data

Tejaswini Deoskar[1], **Christos Christodoulopoulos**[2], **Alexandra Birch**[1], **Mark Steedman**[1]
[1]School of Informatics, University of Edinburgh, Edinburgh, EH8 9AB
[2]University of Illinois, Urbana-Champaign, Urbana, IL 61801
{tdeoskar,abmayne,steedman}@inf.ed.ac.uk, christod@illinois.edu

Abstract

Statistical parsers trained on labeled data suffer from sparsity, both grammatical and lexical. For parsers based on strongly lexicalized grammar formalisms (such as CCG, which has complex lexical categories but simple combinatory rules), the problem of sparsity can be isolated to the lexicon. In this paper, we show that semi-supervised Viterbi-EM can be used to extend the lexicon of a generative CCG parser. By learning complex lexical entries for low-frequency and unseen words from unlabeled data, we obtain improvements over our supervised model for both in-domain (WSJ) and out-of-domain (questions and Wikipedia) data. Our learnt lexicons when used with a discriminative parser such as C&C also significantly improve its performance on unseen words.

1 Introduction

An important open problem in natural language parsing is to generalize supervised parsers, which are trained on hand-labeled data, using unlabeled data. The problem arises because further hand-labeled data in the amounts necessary to significantly improve supervised parsers are very unlikely to be made available. Generalization is also necessary in order to achieve good performance on parsing in textual domains other than the domain of the available labeled data. For example, parsers trained on Wall Street Journal (WSJ) data suffer a fall in accuracy on other domains (Gildea, 2001).

In this paper, we use self-training to generalize the lexicon of a Combinatory Categorial Grammar (CCG) (Steedman, 2000) parser. CCG is a strongly lexicalized formalism, in which every word is associated with a syntactic category (similar to an elementary syntactic structure) indicating its subcategorization potential. Lexical entries are fine-grained and expressive, and contain a large amount of language-specific grammatical information. For parsers based on strongly lexicalized formalisms, the problem of grammar generalization can be cast largely as a problem of lexical extension.

The present paper focuses on learning lexical categories for words that are *unseen* or *low-frequency* in labeled data, from unlabeled data. Since lexical categories in a strongly lexicalized formalism are complex, fine-grained (and far more numerous than simple part-of-speech tags), they are relatively sparse in labeled data. Despite performing at state-of-the-art levels, a major source of error made by CCG parsers is related to unseen and low-frequency words (Hockenmaier, 2003; Clark and Curran, 2007; Thomforde and Steedman, 2011). The unseen words for which we learn categories are surprisingly commonplace words of English; examples are *conquered, apprehended, subdivided, scoring, denotes, hunted, obsessed, residing, migrated* (Wikipedia). Correctly learning to parse the predicate-argument structures associated with such words (expressed as lexical categories in the case of CCG), is important for open-domain parsing, not only for CCG but indeed for any parser.

We show that a simple self-training method, Viterbi-EM (Neal and Hinton, 1998) when used to enhance the lexicon of a strongly-lexicalized parser can be an effective strategy for self-training and domain-adaptation. Our learnt lexicons improve on the lexical category accuracy of two supervised CCG parsers (Hockenmaier (2003) and the Clark and Curran (2007) parser, C&C) on within-domain (WSJ) and out-of-domain test sets (a question corpus and a Wikipedia corpus).

In most prior work, when EM was initialized based on labeled data, its performance did not improve over the supervised model (Merialdo, 1994;

Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 126–134,
Gothenburg, Sweden, April 26-30 2014.

Charniak, 1993). We found that in order for performance to improve, unlabeled data should be used only for parameters which are not well covered by the labeled data, while those that are well covered should remain fixed.

In an additional contribution, we compare two strategies for treating unseen words (a smoothing-based, and a part-of-speech back-off method) and find that a smoothing-based strategy for treating unseen words is more effective for semi-supervised learning than part-of-speech back-off.

2 Combinatory Categorial Grammar

Combinatory Categorial Grammar (CCG) (Steedman, 2000) is a strongly lexicalized grammar formalism, in which the lexicon contains all language-specific grammatical information. The lexical entry of a word consists of a syntactic category which expresses the subcategorization potential of the word, and a semantic interpretation which defines the compositional semantics (Lewis and Steedman, 2013). A small number of combinatory rules are used to combine constituents, and it is straightforward to map syntactic categories to a logical form for semantic interpretation.

For statistical CCG parsers, the lexicon is learnt from labeled data, and is subject to sparsity due to the fine-grained nature of the categories. Figure 1 illustrates this with a simple CCG derivation. In this sentence, *bake* is used as a ditransitive verb and is assigned the ditransitive category **S\NP/NP/NP** . This category defines the verb syntactically as mapping three **NP** arguments to a sentence **S** , and semantically as a ternary relation between its three arguments, thus providing a complete analysis of the sentence.

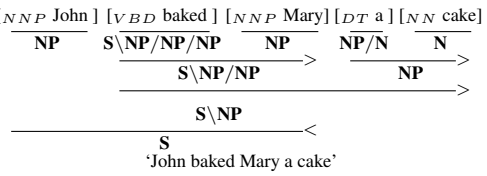

Figure 1: Example CCG derivation

For a CCG parser to obtain the correct derivation above, its lexicon must include the ditransitive category **S\NP/NP/NP** for the verb *bake*. It is not sufficient to have simply seen the verb in another context (say a transitive context like "John baked a cake", which is a more common context). This is in contrast to standard treebank parsers where the verbal category is simply VBD (past tense verb) and a ditransitive analysis of the sentence is not ruled out as a result of the lexical category.

In addition to sparsity related to open-class words like verbs as in the above example, there are also missing categories in labeled data for closed-class words like question words, due to the small number of questions in the Penn Treebank. In general, lexical sparsity for a statistical CCG parser can be broken down into three types: *(i)* where a word is unseen in training data but is present in test data, *(ii)* where a word is seen in the training data but not with the category type required in the test data (but the category type is seen with other words) and *(iii)* where a word bears a category type required in the test data but the category type is completely unseen in the training data.

In this paper, we deal with the first two kinds. The third kind is more prevalent when the size of labeled data is comparatively small (although, even in the case of the English WSJ CCG treebank, there are several attested category types that are entirely missing from the lexicon, Clark et al., 2004). We make the assumption here that all category types in the language have been seen in the labeled data. In principle new category types may be introduced independently without affecting our semi-supervised process (for instance, manually, or via a method that predicts new category types from those seen in labeled data).

3 Related Work

Previous attempts at harnessing unlabeled data to improve supervised CCG models using methods like self-training or co-training have been unsatisfactory (Steedman et al., 2003, 43-44). Steedman et al. (2003) experimented with self-training a generative CCG parser, and co-training a generative parser with an HMM-based supertagger. Co-training (but not self-training) improved the results of the parser when the seed labeled data was small. When the seed data was large (the full treebank), i.e., the supervised baseline was high, co-training and self-training both failed to improve the parser.

More recently, Honnibal et al. (2009) improved the performance of the C&C parser on a domain-adaptation task (adaptation to Wikipedia text) using self-training. Instead of self-training the parsing model, they re-train the supertagging model, which in turn affects parsing accuracy. They obtained an improvement of 1.09% (dependency

score) on supertagger accuracy on Wikipedia (although performance on WSJ text dropped) but did not attempt to re-train the parsing model.

An orthogonal approach for extending a CCG lexicon using unlabeled data is that of Thomforde and Steedman (2011), in which a CCG category for an unknown word is derived from partial parses of sentences with just that one word unknown. The method is capable of inducing unseen categories *types* (the third kind of sparsity mentioned in §2.1), but due to algorithmic and efficiency issues, it did not achieve the broad-coverage needed for grammar generalisation of a high-end parser. It is more relevant for low-resource languages which do not have substantial labeled data and category type discovery is important.

Some notable positive results for non-CCG parsers are McClosky et al. (2006) who use a parser-reranker combination. Koo et al. (2008) and Suzuki et al. (2009) use unsupervised word-clusters as features in a dependency parser to get lexical dependencies. This has some notional similarity to categories, since, like categories, clusters are less fine-grained than words but more fine-grained than POS-tags.

4 Supervised Parser

The CCG parser used in this paper is a re-implementation of the generative parser of Hockenmaier and Steedman (2002) and Hockenmaier (2003)[1], except for the treatment of unseen and low-frequency words.

We use a model (the *LexCat* model in Hockenmaier (2003)) that conditions the generation of constituents in the parse tree on the *lexical category* of the head word of the constituent, but not on the head word itself. While fully-lexicalized models that condition on words (and thus model word-to-word dependencies) are more accurate than unlexicalized ones like the *LexCat* model, we use an unlexicalized model[2] for two reasons: first, our lexicon smoothing procedure (described in the next section) introduces new words and new categories for words into the lexicon. Lexical categories are added to the lexicon for seen and unseen words, but no new category types are introduced. Since the *LexCat* model conditions rule expansions on lexical categories, but not on words, it is still able to produce parses for sentences with new words. In contrast, a fully lexicalized model would need all components of the grammar to be smoothed, a task that is far from trivial due to the resulting explosion in grammar size (and one that we leave for future work).

Second, although lexicalized models perform better on in-domain WSJ data (the *LexCat* model has an accuracy of 87.9% on Section 23, as opposed to 91.03% for the head-lexicalized model in Hockenmaier (2003) and 91.9% for the C&C parser), our parser is more accurate on a question corpus, with a lexical category accuracy of 82.3%, as opposed to 71.6% and 78.6% for the C&C and Hockenmaier (2003) respectively.

4.1 Handling rare and unseen words

Existing CCG parsers (Hockenmaier (2003) and Clark and Curran (2007)) back-off rare and unseen words to their POS tag. The POS-backoff strategy is essentially a pipeline approach, where words are first tagged with coarse tags (POS tags) and finer tags (CCG categories) are later assigned, by the parser (Hockenmaier, 2003) or the supertagger (Clark and Curran, 2007). As POS-taggers are much more accurate than parsers, this strategy has given good performance in general for CCG parsers, but it has the disadvantage that POS-tagging errors are propagated. The parser can never recover from a tagging error, a problem that is serious for words in the Zipfian tail, where these words might also be unseen for the POS tagger and hence more likely to be tagged incorrectly. This issue is in fact more generally relevant than for CCG parsers alone—the dependence of parsers on POS-taggers was cited as one of the problems in domain-adaptation of parsers in the NAACL-2012 shared task on parsing the web (Petrov and McDonald, 2012). Lease and Charniak (2005) obtained an improvement in the accuracy of the Charniak (2000) parser on a biomedical domain simply by training a new POS tagger model.

In the following section, we describe an alternative smoothing-based approach to handling un-

[1] These generative models are similar to the Collins' head-based models (Collins, 1997), where for every node, a head is generated first, and then a sister conditioned on the head. Details of the models are in Hockenmaier and Steedman (2002) and Hockenmaier 2003:pg 166.

[2] A terminological clarification: *unlexicalized* here refers to the model, in the sense that head-word information is not used for rule-expansion. The formalism itself (CCG) is referred to as *strongly-lexicalized*, as used in the title of the paper. Formalisms like CCG and LTAG are considered strongly-lexicalized since linguistic knowledge (functions mapping words to syntactic structures/semantic interpretations) is included in the lexicon.

seen and rare words. This method is less sensitive to POS tagging errors, as described below. In this approach, in a pre-processing step prior to parsing, categories are introduced into the lexicon for unseen and rare words from the data to be parsed. Some probability mass is taken from seen words/categories and given to unseen word and category pairs. Thus, at parse time, no word is unseen for the parser.

4.1.1 Smoothing

In our approach, we introduce lexical entries for words from the unlabeled corpus that are unseen in the labeled data, and also add categories to existing entries for rarely seen words. The most general case of this would be to assign all known categories to a word. However, doing this reduces the lexical category accuracy.[3] A second option, chosen here, is to limit the number of categories assigned to the word by using some information about the word (for instance, its part-of-speech). Based on the part-of-speech of an unseen word in the unlabeled or test corpus, we add an entry to the lexicon of the word with the top n categories that have been seen with that part-of-speech in the labeled data. Each new entry of (w, cat), where w is a word and cat is a CCG category, is associated with a count $c(w, cat)$, obtained as described below. Once all (w, cat) entries are added to the lexicon along with their counts, a probability model $P(w|cat)$ is calculated over the entire lexicon.

Our smoothing method is based on a method used in Deoskar (2008) for smoothing a PCFG lexicon. Eq. 1 and 2 apply it to CCG entries for unseen and rare words. In the first step, an out-of-the-box POS tagger is used to tag the unlabeled or test corpus (we use the C&C tagger). Counts of words and POS-tags $c_{corpus}(w, T)$ are obtained from the tagged corpus. For the CCG lexicon, we ultimately need a count for a word w and a CCG category cat. To get this count, we split the count of a word and POS-tag amongst all categories seen with that tag in the supervised data in the same ratio as the ratio of the categories in the supervised data. In Eq. 1, this ratio is $c_{tb}(cat_T)/c_{tb}(T)$ where $c_{tb}(cat_T)$ is the treebank count of a category cat_T seen with a POS-tag T, and $c_{tb}(T)$ is the marginal count of the tag T in the treebank. This ratio makes a more frequent category type more likely than a rarer one for an unseen word. For example, for unseen verbs, it would make the transitive category more likely than a ditransitive one (since transitives are more frequent than ditransitives). There is an underlying assumption here that relative frequencies of categories and POS-tags in the labeled data are maintained in the unlabeled data, which in fact can be thought of as a prior while estimating from unlabeled data (Deoskar et al., 2012).

$$c_{corpus}(w, cat) = \frac{c_{tb}(cat_T)}{c_{tb}(T)} \cdot c_{corpus}(w, T) \quad (1)$$

Additionally, for seen but low-frequency words, we make use of the existing entry in the lexicon. Thus in a second step, we interpolate the count $c_{corpus}(w, cat)$ of a word and category with the supervised count of the same $c_{tb}(w, cat)$ (if it exists) to give the final smoothed count of a word and category $c_{smooth}(w, cat)$ (Eq. 2).

$$c_{smooth}(w, cat) = \lambda \cdot c_{tb}(w, cat) + (1 - \lambda) \cdot c_{corpus}(w, cat) \quad (2)$$

When this smoothed lexicon is used with a parser, POS-backoff is not necessary since all needed words are now in the lexicon. Lexical entries for words in the parse are determined not by the POS-tag from a tagger, but directly by the parsing model, thus making the parse less susceptible to tagging errors.

5 Semi-supervised Learning

We use Viterbi-EM (Neal and Hinton, 1998) as the self-training method. Viterbi-EM is an alternative to EM where instead of using the model parameters to find a true posterior from unlabeled data, a posterior based on the single maximum-probability (Viterbi) parse is used. Viterbi-EM has been used in various NLP tasks before and often performs better than classic EM (Cohen and Smith, 2010; Goldwater and Johnson, 2005; Spitkovsky et al., 2010). In practice, a given parsing model is used to obtain Viterbi parses of unlabeled sentences. The Viterbi parses are then treated as training data for a new model. This process is iterated until convergence.

Since we are interested in learning the lexicon, we only consider lexical counts from Viterbi parses of the unlabeled sentences. Other parameters of the model are held at their supervised values. We conducted some experiments where we

[3] For instance, we find that assigning all categories to unseen verbs gives a lexical category accuracy of 52.25 %, as opposed to an accuracy of 65.4% by using top 15 categories, which gave us the best results, as reported later in Table 3.

self-trained all components of the parsing model, which is the usual case of self-training. We obtained negative results similar to Steedman et al. (2003), where self-training reduced the performance of the parsing model. We do not report them here. Thus, using unlabeled data only to estimate parameters that are badly estimated from labeled data (lexical entries in CCG, due to lexical sparsity) results in improvements, in contrast to prior work with semi-supervised EM.

As is common in semi-supervised settings, we treated the count of each lexical event as the weighted count of that event in the labeled data (treebank)[4] and the count from the Viterbi-parses of unlabeled data. Here we follow Bacchiani et al. (2006) and McClosky et al. (2006) who show that count merging is more effective than model interpolation.

We placed an additional constraint on the contribution that the unlabeled data makes to the semi-supervised model—we only use counts (from unlabeled data) of lexical events that are rarely seen/unseen in the labeled data. Our reasoning was that many lexical entries are estimated accurately from the treebank (for example, those related to function words and other high-frequency words) and estimation from unlabeled data might hurt them. We thus had a cut-off frequency (of words in labeled data) above which we did not allow the unlabeled counts to affect the semi-supervised model. In practise, our experiments turned out to be fairly insensitive to the value of this parameter, on evaluations over rare or unseen verbs. However, overall accuracy would drop slightly if this cut-off was increased. We experimented with cut-offs of 5, 10 and 15, and found that the most conservative value (of 5) gave the best results on in-domain WSJ experiments, and a higher value of 10 gave the best results for out-of-domain experiments.

We also conducted some limited experiments with classical semi-supervised EM, with similar settings of weighting labeled counts, and using unlabeled counts only for rare/unseen events. Since it is a much more computationally expensive procedure, and most of the results did not come close to the results of Viterbi-EM, we did not pursue it.

[4]The labeled count is weighted in order to scale up the labeled data which is usually smaller in size than the unlabeled data, to avoid swamping the labeled counts with much larger unlabeled counts.

5.1 Data

Labeled: Sec. 02-21 of CCGbank (Hockenmaier and Steedman, 2007). In one experiment, we used Sec. 02-21 minus 1575 sentences that were held out to simulate test data containing unseen verbs—see §6.2 for details.

Unlabeled: For in-domain experiments, we used sentences from the unlabeled WSJ portion of the ACL/DCI corpus (LDC93T1, 1993), and the WSJ portion of the ANC corpus (Reppen et al., 2005), limited to sentences containing 20 words or less, creating datasets of approximately 10, 20 and 40 million words each. Additionally, we have a dataset of 140 million words – 40M WSJ words plus an additional 100M from the New York Times.

For domain-adaptation experiments, we use two different datasets. The first one consists of question-sentences – 1328 unlabeled questions, obtained by removing the manual annotation of the question corpus from Rimell and Clark (2008). The second out-of-domain dataset consists of Wikipedia data, approximately 40 million words in size, with sentence length < 20 words.

5.2 Experimental setup

We ran our semi-supervised method using our parser with a smoothed lexicon (from §4.1.1) as the initial model, on unlabeled data of different sizes/domains. For comparison, we also ran experiments using a POS-backed off parser (the original Hockenmaier and Steedman (2002) *LexCat* model) as the initial model. Viterbi-EM converged at 4-5 iterations. We then parsed various test sets using the semi-supervised lexicons thus obtained. In all experiments, the labeled data was scaled to match the size of the unlabeled data. Thus, the scaling factor of labeled data was 10 for unlabeled data of 10M words, 20 for 20M words, etc.

5.3 Evaluation

We focused our evaluations on unseen and low-frequency verbs, since verbs are the most important open-class lexical entries and the most ambiguous to learn from unlabeled data (approx. 600 categories, versus 150 for nouns). We report lexical category accuracy in parses produced using our semi-supervised lexicon, since it is a direct measure of the effect of the lexicon.[5] We discuss four

[5]Dependency recovery accuracy is also used to evaluate performance of CCG parsers and is correlated with lexical

	All words	All Verbs	Unseen Verbs
SUP	87.76	78.10	52.54
SEMISUP	**88.14**	**78.46**	****57.28**
SUP_{bkoff}	87.91	76.08	54.14
SEMISUP_{bkoff}	87.79	75.68	54.60

Table 1: Lexical category accuracy on TEST-4SEC
**: $p < 0.004$, McNemar test

experiments below. The first two are on in-domain (WSJ) data. The last two are on out-of-domain data – a question corpus and a Wikipedia corpus.

6 Results

6.1 In-domain: WSJ unseen verbs

Our first testset consists of a concatenation of 4 sections of CCGbank (01, 22, 24, 23), a total of 7417 sentences, to form a testset called TEST-4SEC. We use all these sections in order to get a reasonable token count of unseen verbs, which was not possible with Sec. 23 alone.

Table 1 shows the performance of the smoothed supervised model (SUP) and the semi-supervised model (SEMISUP) on this testset. There is a significant improvement in performance on unseen verbs, showing that the semi-supervised model learns good entries for unseen verbs over and above the smoothed entry in the supervised lexicon. This results in an improvement in the overall lexical category accuracy of the parser on all words, and all verbs.

We also performed semi-supervised training using a supervised model that treated unseen words with a POS-backoff strategy SUP_{bkoff}. We used the same settings of cut-off and the same scaling of labeled counts as before. The supervised backed-off model performs somewhat better than the supervised smoothed model. However, it did not improve as much as the smoothed one from unlabeled data. Additionally, the overall accuracy of SEMISUP_{bkoff} fell below the supervised level, in contrast to the smoothed model, where overall numbers improved. This could indicate that the accuracy of a POS tagger on unseen words, especially verbs, may be an important bottleneck in semi-supervised learning.

Low-frequency verbs We also obtain improvements on verbs that are seen but with a low frequency in the labeled data (Table 2). We divided

Freq. Bin	1-5	6-10	11-20
SUP	64.13	75.19	77.6
SEMISUP	**66.72**	**76.21**	**79.8**

Table 2: Seen but rare verbs, TEST-4SEC

verbs occurring in TEST-4SEC into different bins according to their occurrence frequency in the labeled data (bins of frequency 1-5, 6-10 and 11-20). Semi-supervised training improves over the supervised baseline for all bins of low-frequency verbs. Note that our cut-off frequency for using unlabeled data is 5, but there are improvements in the 6-10 and 11-20 bins as well, suggesting that learning better categories for rare words (below the cut-off) impacts the accuracy of words above the cut-off as well, by affecting the rest of the parse positively.

6.2 In-domain : heldout unseen verbs

The previous section showed significant improvement in learning categories for verbs that are unseen in the training sections of CCGbank. However, these verbs are in the Zipfian tail, and for this reason have fairly low occurrence frequencies in the unlabeled corpus. In order to estimate whether our method will give further improvements in the lexical categories for these verbs, we would need unlabeled data of a much larger size. We therefore designed an experimental scenario in which we would be able to get high counts of unseen verbs from a similar size of unlabeled data. We first made a list of N verbs from the treebank and then extracted all sentences containing them (either as verbs or otherwise) from CCGbank training sections. These sentences form a testset of 1575 sentences, called TEST-HOV (for *held out verbs*). The verbs in the list were chosen based on occurrence frequency f in the treebank, choosing all verbs that occurred with a frequency of $f = 11$. This number gave us a large enough set and a good type/token ratio to reliably evaluate and analyze our semi-supervised models—112 verb types, with 1115 token occurrences [6]. Since these verbs are actually mid-frequency verbs in the supervised data, they have a correspondingly large occurrence frequency in the unlabeled data, occurring much more often than true unseen verbs. Thus, the unlabeled data size is effectively magnified—as far as these verbs are concerned, the unlabeled data is approximately 11 times larger than it actually is.

Table 3 shows lexical category accuracy on

category accuracy, but a dependency evaluation is more relevant when comparing performance with parsers in other formalisms and does not have much utility here.

[6] Selecting a different but close value of f such as $f = 10$ or $f = 12$ would have also served this purpose.

	All Words	All Verbs	Unseen Verbs
SUP	87.26	74.55	65.49
SEMISUP	**87.78**	**75.30**	*** **70.43**
SUP_{bkoff}	87.58	73.06	67.25
$SEMISUP_{bkoff}$	87.52	72.89	68.05

Table 3: Lexical category accuracy in TEST-HOV. ***p<0.0001, McNemar test

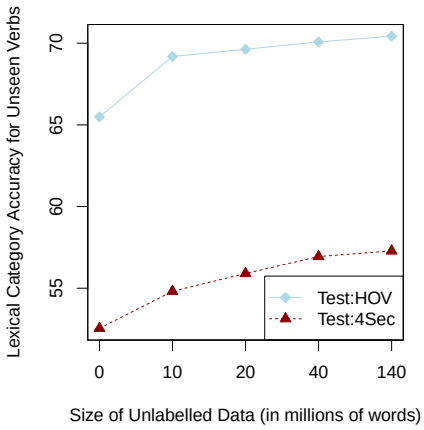

Figure 2: Increasing accuracy on unseen verbs with increasing amounts of unlabeled data.

this testset. The baseline accuracy of the parser on these verbs is much higher than that on the truly unseen verbs.[7] The semi-supervised model (SEMISUP) improves over the supervised model SUP very significantly on these unseen verbs. We also see an overall improvement on all verbs (seen and unseen) in the test data, and in the overall lexical category accuracy as well. Again, the backed-off model does not improve as much as the smoothed model, and moreover, overall performance falls below the supervised level.

Figure 2 shows the effect of different sizes of unlabeled data on accuracy of unseen verbs for the two testsets TEST-HOV and TEST-4SEC . Improvements are monotonic with increasing unlabeled data sizes, up to 40M words. The additional 100M words of NYT also improve the models but to a lesser degree, possibly due to the difference in domain. The graphs indicate that the method will lead to more improvements as more unlabeled data (especially WSJ data) is added.

[7]This could be because verbs in the Zipfian tail have more idiosyncratic subcategorization patterns than mid-frequency verbs, and thus are harder for a parser. Another reason is that they may have been seen as nouns or other parts of speech, leading to greater ambiguity in their case.

	QUESTIONS		WIKIPEDIA	
	All words	wh words	All words	Unseen words
SUP	82.36	61.77	84.31	79.5
SEMISUP	***83.21**	**63.22**	***85.6**	**80.25**

Table 4: Out-of-domain: Questions and Wikipedia, *p<0.05, McNemar test

6.2.1 Out-of-Domain

Questions The question corpus is not strictly a different domain (since questions form a different kind of construction rather than a different domain), but it is an interesting case of adaptation for several reasons: WSJ parsers perform poorly on questions due to the small number of questions in the Penn Treebank/CCGbank. Secondly, unsupervised adaptation to questions has not been attempted before for CCG (Rimell and Clark (2008) did supervised adaptation of their supertagger).

The supervised model SUP already performs at state-of-the-art on this corpus, on both overall scores and on wh(question)-words alone. C&C and Hockenmaier (2003) get 71.6 and 78.6% overall accuracies respectively, and only 33.6 and 50.7 on wh-words alone. To our original unlabeled WSJ data (40M words), we add 1328 unlabeled question-sentences from Rimell and Clark, 2008, scaled by ten, so that each is counted ten times. We then evaluated on a testset containing questions (500 question sentences, from Rimell and Clark (2008)). The overall lexical category accuracy on this testset improves significantly as a result of the semi-supervised learning (Table 4). The accuracy on the question words alone *(who, what, where, when, which, how, whose, whom)* also improves numerically, but by a small amount (the number of tokens that improve are only 7). This could be an effect of the small size of the testset (500 sentences, i.e. 500 wh-words).

Wikipedia We obtain statistically significant improvements in overall scores over a testset consisting of Wikipedia sentences hand-annotated with CCG categories (from Honnibal et al. (2009)) (Table 4). We also obtained improvements in lexical category accuracy on unseen words, and on unseen verbs alone (not shown), but could not prove significance. This testset contains only 200 sentences, and counts for unseen words are too small for significance tests, although there are numeric improvements. However, the overall improvement is statistically significantly, showing that adapting the lexicon alone is effective for a new domain.

6.3 Using semi-supervised lexicons with the C&C parser

To show that the learnt lexical entries may be useful to parsers other than our own, we incorporate our semi-supervised lexical entries into the C&C parser to see if it benefits performance. We do this in a naive manner, as a proof of concept, making no attempt to optimize the performance of the C&C parser (since we do not have access to its internal workings). We take all entries of unseen words from our best semi-supervised lexicon (word, category and count) and add them to the dictionary of the C&C supertagger (tagdict). The C&C is a discriminative, lexicalized model that is more accurate than an unlexicalized model. Even so, the lexical entries that we learn improve the C&C parsers performance over and above its back-off strategy for unseen words. Table 5 shows the results on WSJ data TEST-4SEC and TEST-HOV. There were numeric improvements on the TEST-4SEC test set as shown in Table 5[8]. We obtain significance on the TEST-HOV testset which has a larger number of tokens of unseen verbs and entries that were learnt from effectively larger unlabeled data. We tested two cases: when these verbs were seen for the POS tagger used to tag the test data, and when they were unseen for the POS tagger, and found statistically significant improvement for the case when the verbs were unseen for the POS tagger[9], indicating sensitivity to POS-tagger errors.

	TEST-4SEC	TEST-HOV	
	(590)	POS-seen (1134)	POS-unseen (1134)
C&C	62.03 (366)	76.71 (870)	72.39 (821)
C&C (enhanced)	**63.89** (377)	**77.34** (877)	***73.98** (839)

Table 5: TEST-4SEC: Lexical category accuracy of C&C parser on unseen verbs. Numbers in brackets are the number of tokens.*p<0.05, McNemar test

6.4 Entropy and KL-divergence

We also evaluated the quality of the semi-supervised lexical entries by measuring the overall entropy and the average Kullback-Leibler (KL) divergence of the learnt entries of unseen verbs from entries in the gold testset. The gold entry for each verb from the TEST-HOV testset was obtained from the heldout gold treebank trees. Supervised (smoothed) and semi-supervised entries were obtained from the respective lexicons. These metrics use the conditional probability of a category given a word, which is not a factor in the generative model (which considers probabilities of words given categories), but provide a good measure of how close the learnt lexicons are to the gold lexicon. We find that the average KL divergence reduces from **2.17** for the baseline supervised entries to **1.40** for the semi-supervised entries. The overall entropy for unseen verb distributions also goes down from **2.23** (supervised) to **1.37** (semi-supervised), showing that semi-supervised distributions are more peaked, and bringing them closer to the true entropy of the gold distribution (**0.93**).

7 Conclusions

We have shown that it is possible to learn CCG lexical entries for unseen and low-frequency words from unlabeled data. When restricted to learning only lexical entries, Viterbi-EM improved the performance of the supervised parser (both in-domain and out-of-domain). Updating all parameters of the parsing model resulted in a decrease in the accuracy of the parser. We showed that the entries we learnt with an unlexicalized model were accurate enough to also be useful to a highly-accurate lexicalized parser. It is likely that a lexicalized parser will provide even better lexical entries. The lexical entries continued to improve with increasing size of unlabeled data. For the out-of-domain testsets, we obtained statistically significant overall improvements, but we were hampered by the small sizes of the testsets in evaluating unseen/wh words.

In future work, we would like to add unseen but predicted category *types* to the initial lexicon using an independent method, and then apply the same semi-supervised learning to words of these types.

Acknowledgements

We thank Mike Lewis, Shay Cohen and the three anonymous EACL reviewers for helpful comments. This work was supported by the ERC Advanced Fellowship 249520 GRAMPLUS.

[8]There were also improvements on the question and Wikipedia testsets (not shown) (8 and 6 tokens each) but the size of these testsets is too small for significance.

[9]Note that for this testset TEST-HOV, the numbers are the supertagger's accuracy, and not the parser's. We were only able to retrain the supertagger on training data with TEST-HOV sentences heldout, but could not retrain the parser, despite consultation with the authors.

References

Michiel Bacchiani, Michael Riley, Brian Roark, and Richard Sproat. 2006. MAP adaptation of stochastic grammars. *Computer Speech and Language*, 20(1):41–68.

Eugene Charniak. 1993. *Statistical Language Learning*. MIT Press.

Stephen Clark and James R. Curran. 2007. Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models. *Computational Linguistics*, 33(4):493–552.

Stephen Clark, Mark Steedman, and James Curran. 2004. Object-extraction and question-parsing using CCG. In *Proceedings of EMNLP 2004*.

Shay Cohen and Noah Smith. 2010. Viterbi Training for PCFGs: Hardness Results and Competitiveness of Uniform Initialization. In *Proceedings of ACL 2010*.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th ACL*.

Tejaswini Deoskar. 2008. Re-estimation of Lexical Parameters for Treebank PCFGs. In *Proceedings of COLING 2008*.

Tejaswini Deoskar, Markos Mylonakis, and Khalil Sima'an. 2012. Learning Structural Dependencies of Words in the Zipfian Tail. *Journal of Logic and Computation*.

Daniel Gildea. 2001. Corpus Variation and Parser Performance. In *Proceedings of EMNLP 2001*.

Sharon Goldwater and Mark Johnson. 2005. Bias in learning syllable structure. In *Proceedings of CoNLL05*.

Julia Hockenmaier. 2003. *Data and Models for Statistical Parsing with Combinatory Categorial Grammar*. Ph.D. thesis, School of Informatics, University of Edinburgh.

Julia Hockenmaier and Mark Steedman. 2002. Generative Models for Statistical Parsing with Combinatory Categorial Grammar. In *ACL40*.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank. *Computational Linguistics*, 33:355–396.

Matthew Honnibal, Joel Nothman, and James R. Curran. 2009. Evaluating a Statistial CCG Parser on Wikipedia. In *Proceedings of the 2009 Workshop on the People's Web Meets NLP, ACL-IJCNLP*.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple Semi-supervised Dependency Parsing. In *Proceedings of ACL-08: HLT*, pages 595–603. Association for Computational Linguistics, Columbus, Ohio.

LDC93T1. 1993. LDC93T1. *Linguistic Data Consortium, Philadelphia*.

Matthew Lease and Eugene Charniak. 2005. Parsing Biomedical Literature. In R. Dale, K.-F. Wong, J. Su, and O. Kwong, eds., *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP'05)*, vol. 3651 of *Lecture Notes in Computer Science*, pages 58 – 69. Springer-Verlag, Jeju Island, Korea.

Mike Lewis and Mark Steedman. 2013. Combined Distributional and Logical Semantics. *Transactions of the Association for Computational Linguistics*.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective Self-Training for Parsing. In *Proceedings of HLT-NAACL 2006*.

Bernard Merialdo. 1994. Tagging English Text with a Probabilistic Model. *Computational Linguistics*, 20(2):155–171.

Radford M. Neal and Geoffrey E. Hinton. 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning and Graphical Models*, pages 355 – 368. Kluwer Academic Publishers.

Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 Shared Task on Parsing the Web. In *First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL) Workshop at NAACL 2012*.

Randi Reppen, Nancy Ide, and Keith Suderman. 2005. LDC2005T35, American National Corpus (ANC) Second Release. *Linguistic Data Consortium, Philadelphia*.

Laura Rimell and Stephen Clark. 2008. Adapting a Lexicalized-Grammar Parser to Contrasting Domains. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-08)*.

Valentin I. Spitkovsky, Hiyan Alshawi, Daniel Jurafsky, and Christopher D. Manning. 2010. Viterbi Training Improves Unsupervised Dependency Parsing. In *Proceedings of CoNLL-2010*.

Mark Steedman. 2000. *The Syntactic Process*. MIT Press/Bradford Books.

Mark Steedman, Steven Baker, Jeremiah Crim, Stephen Clark, Julia Hockenmaier, Rebecca Hwa, Miles Osbornn, Paul Ruhlen, and Anoop Sarkar. 2003. Semi-Supervised Training for Statistical Parsing. Tech. rep., CLSP WS-02.

Jun Suzuki, Hideki Isozaki, Xavier Carreras, and Michael Collins. 2009. An Empirical Study of Semi-supervised Structured Conditional Models for Dependency Parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 551–560. Association for Computational Linguistics, Singapore.

Emily Thomforde and Mark Steedman. 2011. Semi-supervised CCG Lexicon Extension. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, Edinburgh UK*.