# Report

## Using Combinatory Categorial Grammar in the Structured Language Model

Peng Xu and Frederick Jelinek
CLSP/Electrical and Computer Engineering

**AIMS** We study the impact of using combinatory categorial grammar (CCG) as a natural enrichment of the syntactical labels in the structured language model (SLM). Perplexity (PPL) and word-error-rate (WER) from N-best rescoring will be used as performance measures.

**BACKGROUND** CCG is a wide-converage parsing technique that has the potential benefit of more constrained grammar and simple and semantically transparent capture of extraction and coordination [1]. CCG grammars have much larger category sets than standard Penn Treebank grammars that we used in our previous SLM studies [2, 3]. For example, CCG grammars distinguish between many classes of verbs with different subcategorization frames. As a result of simple unary and binary combinatory schemata such as function application and composition, CCG has a smaller and less overgenerating grammer than standard PCFGs.

In CCG, most aspects of the grammar are specified in the categories of the lexical items, identifying a lexical item as either a *functor* or *argument*. For example, the transitive verb **bought** can have the following category that specifies its first argument as a noun phrase (NP) to its right and its second argument as an NP to its left, and its result as a sentence (S):

```
bought := (S\NP)/NP
```

In this notation, the NP after "/" is the argument to the right, the NP after "\" is the argument to the left. For details about applying combinatory rules, such as functional application and composition, see [4].

Our interest in using CCG in SLM lies in the fact that CCG categories can serve as a natural enrichment of the syntactic information of a lexical item or a constituent in the parse tree, as parsing goes along. Our previous study [5, 3] showed that richer syntactic dependencies lead to improvements in both PPL and WER, when applied to the SLM. Since CCG categories are context dependent, we will have a context dependent enrichment of the syntactic heads, as opposed to uniform enrichment in our previous study. We would like to see the impact of this enriching scheme on the performance of the SLM. Therefore, in our experiments described below, CCG categories are not enriched by our previously reported schemes.

**EXPERIMENTS & RESULTS** We evaluated the PPL performance of the CCG style SLM (CCG-SLM) on the UPenn Treebank data. In order to initialize the CCG-SLM, we have to use parse trees with CCG annotation. CCGbank is a corpus of CCG normal-form derivations obtained by translating the UPenn Treebank trees, as described in [6]. The derivations in CCGbank are in "normal-form" because analyses involving the combinatory rules of type-raising and composition are only used when syntactically necessary.[1]

We partitioned the CCGbank into three parts: training (section 00-20), heldout (section 21-22) and test (section 23-24). Three iterations of EM training were carried out and PPL results are shown in Table 1.

| Model | Iter | PPL | Reduction |
|---|---|---|---|
| 3-gram | | 166.6 | - |
| 3-gram(CCG) | | 173.1 | - |
| h-2+OP+PA | 0 | 144.1 | 13.5% |
| h-2+OP+PA | 3 | 143.8 | 13.7% |
| CCG-SLM | 0 | 147.7 | 14.7% |
| CCG-SLM | 3 | 147.0 | 15.1% |

Table 1: PPL results

[1]Thanks to Julia Hockenmaier who kindly provided us the first version of the CCGbank.

In Table 1, we have 3-gram as our baseline and we intend to compare the CCG-SLM to our best previous enriching scheme *h-2+OP+PA* of the standard SLM as reported in [3]. However, since CCGbank was used to initialize the CCG-SLM models and there are differences in tokenization and regularization, we have two different 3-gram results. In order to have a fair comparison, we included the relative PPL reduction from corresponding 3-gram in the table. It is clear that the CCG-SLM achieved more reduction in PPL than the best result we reported previously.

We also evaluated the CCG-SLM as a second-pass language model for speech recognition. The standard WSJ DARPA'93 HUB1 setup was used as the test-bed and N-best rescoring was performed with CCG-SLM. To initialize CCG-SLM from a reasonable amount of parsed data, we used the CCG-SLM trained on CCGbank as a parser to parse 20M words of WSJ text (the same as we used in [3]). The parsed data then was used to train the final CCG-SLM as a language model. Table 2 shows the WER when CCG-SLM is interpolated with the 3-gram with different interpolation weights. For comparison, we again include results from the enriching scheme *h-2+OP+PA* in the table.

| Model | $\lambda$ | | | | | |
|---|---|---|---|---|---|---|
| | 0.0 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
| h-2+OP+PA | 12.6 | 12.4 | 12.5 | 12.7 | 12.9 | 13.7 |
| CCG-SLM | 13.4 | 13.3 | 13.3 | 12.9 | 13.3 | 13.7 |

Table 2: N-best re-scoring WER(%) results

**ANALYSIS** Our results show that using CCG in the SLM gave improvement in PPL, over the best result we reported previously, but in experiments on N-best rescoring, CCG-SLM did not achieve satisfactory results. However, as opposed to previous experiments, the data we used in WER experiments were a lot more noisy. The SLM, as a parser, has a much worse parsing performance compared to other state-of-the-art parsers [3]. Therefore, the WER results we got with CCG-SLM is an under-estimate of the real power it has.

In all the experiments mentioned in this report, we used linear interpolation as the smoothing method in all models. Although this may be a fairly good choice in the original SLM, linear interpolation is not likely to be as good for the CCG-SLM because of the large number of CCG categories we are using. As alternatives, other smoothing techniques should be studied under CCG-SLM framework, e.g., history clustering, maximum entropy, etc.

# References

[1] Julia Hockenmaier and Mark Steedman, "Generative models for statistical parsing with Combinatory Categorial Grammar," in *Proceedings of the 40th Annual Meeting of the ACL*, pp. 335–342, Philadelphia, PA. July 2002.

[2] Ciprian Chelba and Frederick Jelinek, "Structured language modeling," in*Computer Speech and Language*, 14(4):283–332, October 2000.

[3] Peng Xu, Ciprian Chelba and Frederick Jelinek, "A study on richer syntactic dependencies for structured language modeling," in *Proceedings of the 40th Annual Meeting of the ACL*, pp. 191–198, Philadelphia, PA. July 2002.

[4] Mark Steedman, "The Syntactic Process." The MIT Press, Cambridge, MA. 2000.

[5] Ciprian Chelba and Peng Xu, "Richer syntactic dependencies for structured language modeling," in *Proceedings of the Automatic Speech Recognition and Understanding Workshop*, Madonna di Campiglio, Trento-Italy, December, 2001.

[6] Julia Hockenmaier and Mark Steedman, "Acquiring compact lexicalized grammars from a cleaner treebank," in *Proceedings of the Third LREC Conference*, Las Palmas, Spain, 2002.