Generative CCG Parsing with OOV Prediction

Huijia Wu
Institute of Automation, Chinese Academy of Science
huijia.wul@ia.ac.cn

Abstract

This paper presents our system for the CIPS-SIGHAN-2014 bakeoff task of Simplified Chinese Parsing (Task 3). The system adopts a generative model with OOV prediction model. The former has a PCFG form while the latter uses a three-layer hierarchical Bayesian model. The final performance on the test corpus is reported together with the performance of the OOV model.

1 Introduction

Statistical parsing is the process of discovering the syntactic relations in a sentence, according to the rules of a formal grammar. There exist a body of parsers based on various linguistic formalisms, such as LFG, HPSG, TAG and CCG. (Riezler et al., 2002; Sarkar and Joshi, 2003; Cahill et al., 2004; Miyao and Tsujii, 2005; Clark and Curran, 2007). The parsing techniques also vary from the generative model to the discriminative model. The former uses a joint probability distribution including both the observations and the targets, while the latter only models the conditional probability measure to describe the randomness of the targets based on the observations (Hockenmaier, 2003a, 2003b; Clark and Curran, 2007).

The out-of-vocabulary (OOV) problem is far-from solved in statistical parsing, especially in CCG. There are lots of categories such that a computer would be less likely to remember a word. Clark proposed a supertagger to assignment several possible categories to a word which provides highly accurate and efficient results (Clark, 2002).

In this task we propose a three layer hierarchical Bayesian model to predict the OOV, using the POS tag as the hidden layer. Further, we estimate a OOV's category through integrating all possible POS tags, which means that we need to find relations between OOV and POS. To achieve this goal, we create a mapping between a CCG tree and a TCT tree, which is another kind of syntactic tree according the Tsinghua Chinese Treebank (TCT).

Leaf nodes	Unary trees	Head left:	Head right:
(S\NP)/NP 喜欢	S/(S\NP) NP	S\NP (S\NP)/NP NP	S NP S\NP

Table 1: The four different kinds of expansion

The final report has two parts, one is the evaluation performance based on the test corpus, the other is the performance on OOV prediction.

2 Our System

Our system combines a generative model for parsing with a OOV prediction model. The former follows heavily from (Hockenmaier, 2003a) with slightly modification, which includes the definition of head nodes, using Dirichlet prior as the smoothing technique. The latter is a three-layer hierarchical Bayesian model: the input and the output layer corresponds to a OOV and its category, respectively, composed with a POS tag as the hidden layer.

2.1 Generative Model for Parsing

In this evaluation task, we adopt a generative model as the CCG parsing algorithm. One advantage of the generative model is it needs less human intervention than the discriminative model, which means that, if we have enough data, together with the proper generative model, the algorithm can learn from the data, of the data and for the data with a competitive performance, while the discriminative model needs a lot of manual feature templates, which sounds like cheating since the features are designed by human, rather than the computer itself.

Our generative model bases on (Hockenmaier

Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, pages 153–156,
Wuhan, China, 20-21 October 2014

, 2003a), which defines a generative model over CCG derivation trees. This model acts like a PCFG form, which does not incorporate the notion of combinatory trees. Instead, it is a generative model over sub-trees. By contrast to Hockenmaier, we use a different approach of defining head node, which is a functor categories (categories that accept arguments). Since from a modelling point of view, isolating a head node from a non-head one just make a generative process more hierarchical, there is no statistically significant differences between a head node and a non-head node.

The derivations of a CCG tree can be represented by top-down expansions. As mentioned in (Hockenmaier, 2003a), there are four kinds of leaf nodes in a CCG tree, which corresponds to four kinds of expansion (Table 1). Follow this convention, we have the following generating process:

1. **Expansion probabiltiy**: Start from a root, choose a type of expansion N by $P(\text{exp}|C)$ with exp $\in$ {left, right, unary, leaf} and $C \in \mathcal{C}$.

2. **Lexical probability**: If it meets a leaf node, a word w is generated with probability $P(w|C, \text{exp} = \text{leaf})$, stop.

3. **Head probability**: Otherwise, choose a head node with probability $P(H|C, \text{exp})$.

4. **Non-head probability**: Finally, generate a non-head node w.p. $P(D|C, \text{exp}, H)$.

2.2 Inference and Learning

The parameter estimation step is similar to a PCFG parser based on the maximum likelihood estimation (MLE), but the estimator may become sparsity due to the huge number of parameters. This may cause the problem of overestimation. To avoid this, we can use a regularization term or a prior as the smoothing technique.

In this task, we prefer a Dirichlet distribution as the prior to other smoothing methods. Since it is easy to implement and forms a conjugate prior to a multinomial distribution. We put a Dirichlet prior $\text{Dir}(\alpha)$ on a lexical distribution $P(w|C, \text{exp} = \text{leaf})$. In the experiment we set the $\alpha = (1, 1, \ldots, 1)$ as a uniform distribution.

The learning or decoding algorithm is the well-known CKY algorithm. But efficiency is still a problem, since the number of categories is large, for a long sentence more computing steps will be needed to compose two adjacent cells in a chart than other lexicon-based parsers. Fortunately, Clark and Curran proposed an log-likelihood CCG parser which is efficient enough to large-scale NLP tasks (Clark and Curran, 2007).

2.3 Estimating the OOV

The supertagger proposed by Clark uses a maximum entropy model to predict a word's categories, based on the idea that given a set of manual features, we need to find a category distribution restricted on the set acts an uniform predictor to unknown words. This maximum entropy principle may not apply to OOV estimating, for the reasons that the OOV is rare, statistically insignificant and unable to catch by a statistical model.

Manual rules can get a more accurate prediction than the statistical model, but these rules are also non-flexible, time-confusing and heavy-lifting. To overcome this problem, we propose a mapping between a CCG tree and a TCT tree with the same terminal nodes.

To make this mapping possible we first need to verify the existence, uniqueness and reversibility of such a mapping. Luckily such a mapping is exist since the CCG tree is generated by a TCT tree. To make it simpler we omit the condition of the uniqueness and reversibility. Now the problem is: Can we find a such a mapping to help us to predict the OOV?

Obviously, the mapping is the relation between the syntactic symbols (POS) and the semantic symbols (category). If we can find the estimator of $P(\text{cat}|\text{pos})$ our problem is easily solved by:

$$P(O|C) = \frac{P(C|O)P(O)}{\sum_{O \in \{OOV\}} P(C|O)P(O)} \tag{1}$$

$$= \frac{P(O)\sum_{S \in \{POS\}} P(C|S)P(S|O)}{\sum_{O \in \{OOV\}} \left\{ P(O)\sum_{S \in \{POS\}} P(C|S)P(S|O) \right\}} \tag{2}$$

In the above equations, O stands for the OOV, which is a random variable assigned values from all possible OOV. C indicates the category and S stands for POS tag.

How to create such a mapping matrix? We start from the root node, using a depth-first search algorithm to find the correspondence between nodes in each tree. Notice that the CCG tree is binary, while the TCT tree is not. To find the correct map, we

first need to binarize the TCT tree. But the set of all possible binary trees may become huge when there are many children of a node. Fortunately we just need to expand all binary nodes through one direction.

This model acts like the maximum entropy model, since they all use the context features, but the difference is the former focuses on a more restricted conditions based on the tree structure, while the features in the latter is at the sentence level.

3 Experiment

3.1 Datesets

The data uses in the system composed of two parts, one is for the parser, the other is for the OOV prediction model. The data used by the former comes from the sponsor (CCG bank) with 17558 parsed sentences, 984 categories, while the latter uses data from both the CCG bank and the TCT bank with 9034 sentences. To find the mapping tree with the same leaf nodes, we extract such tree pairs from the two data sets. Finally we get a data set for the OOV prediction model with 5360 tree pairs.

3.2 Experimental Results

There are two kinds of metrics to be evaluated, one is the syntactic category evaluation metrics, the other is the parsing tree evaluation metrics. We report both of these metrics, together with the performance of the OOV prediction model.

Table 2 and 3 gives the performance of the parser on the test set, based on the syntactic category evaluation metrics and the parsing tree evaluation metrics, respectively.

The notations in Table 3 are explained as follows (Qiang Zhou, 2014):

- LDP_CE stands for the lexical dependency pairs (LDPs) with complex event relations in the sentence levels.
- LDP_CC stands for the LDPs with concept compound relations in the chunk levels.
- LDP_PA stands for the LDPs with predicate-argument relations in the clause levels, including head-complement and adjunct-head relations.
- LDP_MO stands for the LDPs with other non-PA relations in the chunk and clause levels, including modifier-head and operator-complement relations.

Category	Precision	Recall	F1
NP	79.71	89.07	84.13
NP/NP	63.31	67.63	65.4
Others	70.57	67.47	68.99
All	71.80	71.81	71.81

Table 2: The performance based on the syntactic category evaluation metrics

Relation	Precision	Recall	F1
LDP_CE	12.98	11.92	12.43
LDP_CC	26.80	36.87	31.04
LDP_PA	40.69	40.47	40.58
LDP_MO	45.99	45.33	45.66
Others	45.81	43.62	44.69
All	42.31	42.27	42.29

Table 3: The performance based on the parsing tree evaluation metrics

Table 4 shows the performance of the OOV estimation model, OOV-POS is the baseline model, which means that a node's category is taken exactly on the corresponding POS tag, +head means such a category is not just on its POS tag, but also with its parent's node's POS tag. +sister has the similar meaning.

Model	Precision	Recall	F1
OOV-POS	60.02	72.10	65.46
+parent	83.15	88.12	85.56
+sister	76.2	82.41	79.18
+parent, sister	86.67	90.2	88.39

Table 4: The results of OOV prediction model

4 Conclusion

This report has shown a generative CCG parser with a OOV prediction model. One contribution of this report is the development of a Bayesian model to predict the OOV with high accuracy. The techniques we use is easy to extend to a more complicated system.

Acknowledge

We would like to thank Qiang Zhou for helpful discussion.

References

Clark, Stephen. 2002. *A maximum-entropy-inspired parser* . In Proceedings of the 1st Meeting of the NAACL, pages 132–139, Seattle, WA.

Clark, Stephen and James R. Curran. 2003. *Log-linear models for wide-coverage CCG parsing.*. In Proceedings of the EMNLP Conference, pages 97–104, Sapporo, Japan.

Clark, Stephen and James R. Curran. 2004b. *Parsing the WSJ using CCG and log-linear models*. In Proceedings of the 42nd Meeting of the ACL, pages 104–111, Barcelona, Spain.

Clark, Stephen, Julia Hockenmaier, and Mark Steedman. 2002. *Building deep dependency structures with a wide-coverage CCG parser.* In Proceedings of the 40th Meeting of the ACL, pages 327–334, Philadelphia, PA.

Geman, Stuart and Mark Johnson. 2002. *Dynamic programming for parsing and estimation of stochastic unification-based grammars*. In Proceedings of the 40th Meeting of the ACL, pages 279–286, Philadelphia, PA.

Collins, Michael. 1996. *A new statistical parser based on bigram lexical dependencies*. In Proceedings of the 34th Meeting of the ACL, pages 184–191, Santa Cruz, CA.

Collins, Michael. 1999. *Head-driven statistical models for natural language parsing*. PhD thesis, University of Pennsylvania.

Collins, Michael. 2003. *Head-driven statistical models for natural language parsing*. Computational Linguistics, 29(4):589–637.

Hockenmaier, Julia and Mark Steedman. 2002a *Acquiring compact lexicalized grammars from a cleaner treebank*. In Proceedings of the Third LREC Conference, pages 1974–1981, Las Palmas, Spain.

Hockenmaier, Julia and Mark Steedman. 2002b *Generative models for statistical parsing with Combinatory Categorial Grammar*. In Proceedings of the 40th Meeting of the ACL, pages 335–342, Philadelphia, PA.

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

Hockenmaier, Julia. 2003b *Parsing with generative models of predicate-argument structure*. In Proceedings of the 41st Meeting of the ACL, pages 359–366, Sapporo, Japan.

Lari, K. and S. J. Young. 1990. *The estimation of stochastic context-free grammars using the inside-outside algorithm*. Computer Speech and Language, 4(1):35–56.

Mark Johnson, Thomas L. Griffiths and Sharon Goldwater. 2007. *Inference for PCFGs via Markov Chain Monte Carlo*. Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Confer- ence, pages 139-146.

Percy Liang, Slav Petrov, Michael I. Jordan, Dan Klein. 2007. *The infinite PCFG using hierarchical Dirichlet processes*. Empirical Methods in Natural Language Processing and Computational Natural Language Learning(EMNLP/CoNLL).

Qiang Zhou. 2004. *Chinese Treebank Annotation Scheme*. Journal of Chinese Information, 18(4), p1-8.

Qiang Zhou. 2011. *Automatically transform the TCT data into a CCG bank: designation specification Ver 3.0*. Technical Report CSLT-20110512, Center for speech and language technology, Research Institute of Information Technology, Tsinghua University.

Ratnaparkhi, Adwait, Salim Roukos, and Todd Ward. 1994. *A maximum entropy model for parsing*. In Proceedings of the International Conference on Spoken Language Processing, pages 803–806, Yokohama, Japan.

Sarkar, A. and Joshi, A. 2003. *Tree-adjoining grammars and its application to statistical parsing*. In Bod, R., Scha, R., and Sima'an, K., editors, Data-oriented parsing. CSLI.

Steedman, Mark. 1996. *Surface Structure and Interpretation*. MIT Press, Cambridge, MA.

Steedman, Mark. 2000. *The Syntactic Process*. The MIT Press, Cambridge, MA.

Steedman, Mark, Steven Baker, Stephen Clark, Jeremiah Crim, Julia Hockenmaier, Rebecca Hwa, Miles Osborne, Paul Ruhlen, and Anoop Sarkar. 2002. *Semi-supervised training for statistical parsing: Final report*. Technical report, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD.

Yonatan Bisk and Julia Hockenmaier. 2013. *An HDP Model for Inducing Com- binatory Categorial Grammars*. Transactions of the Association for Compu- tational Linguistics Vol 1.