Supervised Induction of Probabilistic Tree Adjoining Grammars

Sarath Chandar A P APSARATHCHANDAR@GMAIL.COM
Ravindran B RAVI@CSE.IITM.AC.IN
Indian Institute of Technology Madras, Chennai, India, 600036.

Abstract

We propose a Bayesian non-parametric model to induce Tree Adjoining Grammars(TAGs) from natural language based on (Cohn et al., 2010). Tree Adjoining Grammars belong to the family of Mildly Context Sensitive Grammar formalisms(MCSGs) that are rich enough to capture the underlying intricacies of the language. To the best of our knowledge, this is the first work that uses an MCSG formalism coupled with a Bayesian nonparametric model.

1. Introduction

Inducing grammar from natural language, even though extremely challenging, is an active area of research in computational linguistics. Context Free Grammars(CFG) and Probabilistic CFGs (PCFG) are used in the literature for decades due to their simplicity. But a natural language like English is known to be strictly in the class of Context Sensitive Languages (CSLs) which are hard to learn due to their complexity. Thus there is a trade off between learnablity and complexity of the grammar formalism. In an attempt to address this tradeoff (Joshi & Schabes, 1997) proposed Tree Adjoining Grammars(TAGs). TAG belongs to a class of languages called Mildly Context Sensitive Languages (MCSL). TAGs consists of set of inital trees and auxiliary trees. Even though MCSL formalisms like TAG are less complex than CSLs, learning TAG from the natural language is highly challenging, since in TAG induction we need to induce the structure of elementary trees from the data set. One solution to find the balance between learnability and complexity is to use a rich grammar formalism with a nonparametric Bayesian prior to limit the model complexity (Cohn et al., 2010). The advantage of using nonparametric Bayesian prior is that they provide us infinitely many parameters that increases the complexity of the model, provided there is sufficient data. In this paper, we propose a model that uses Tree Adjoining Grammars as the grammar representation, along with a nonparametric Bayesian prior. The experimental results reported shows that this results in the increase in accuracy of parsing.

Appearing in Proceedings of the 1st Indian Workshop on Machine Learning, IIT Kanpur, India, 2013.

2. Proposed Model

Probabilistic Tree Adjoining Grammars (PTAGs), assigns a probability to each rule in the grammar, denoted by $P(e|c)$, where the elementary tree e rewrites the non-terminal c. In this case, e can be either an initial tree or an auxiliary tree. Similar to (Cohn et al., 2010), instead of inferring the grammar from the treebank directly, we infer a distribution over the derivation used to produce the tree. Then we can read the grammar off the elementary trees. This reduces the problem to inferring posterior distribution of e given w.

The probability of a derivation e (denoted by $P(e)$) is the product of the probabilities of its component rules.

$$P(e) = \prod_{c \to e \in (I \cup A)} P(e|c)$$

where $e = (e_1, e_2...)$ is a sequence of initial/auxiliary trees used for the derivation and $c = root(e)$ is the root symbol of e and $c \to e$ means we replace node c with the elementary tree e. We assume that e is conditionally independent of the remaining part of the tree given the root c. We do not know how many initial trees and auxiliary trees are needed to account for the data. So as mentioned earlier, we use Bayesian nonparametric priors that support an infinite distribution over all possible initial and auxiliary trees. We use Pitman-Yor process (PYP), which is a generalization of the Dirichlet Process (DP). For each non-terminal c, we have two PYPs, one process that generates initial trees rooted with c and another process that generates

auxiliary trees rooted with c. Once we have two such distributions, we can decouple the decision about the kind of tree to be used in the derivation from the selection of tree. Assuming we have the base distribution from which we can sample new elementary trees, the derivation $e = e_1, e_2, .., e_n$ can be defined as follows.

$$G_{c_x}|a_{c_x}, b_{c_x}, P_{E_x} \sim PYP(a_{c_x}, b_{c_x}, P_{E_x}(.|c))$$

$$e_{i_x}|c, G_{c_x} \sim G_{c_x}$$

where $x \in \{in, au\}$. $x = in$ denotes that the process generates initial trees and $x = au$ denotes that the process generates auxiliary trees. For example $G_{c_{in}}$ is the distribution over all initial trees with root non-terminal c. G is an infinite distribution over possible elementary trees drawn from the PYP prior. Even though e_i are drawn i.i.d from G, we integrate over possible values of G to introduce dependencies between the e_i. This can be better understood by the variant of Chinese Restaurant Process (CRP) called Pitman-Yor CRP(PYCRP) defined in (Cohn et al., 2010).

PYCRP consists of a restaurant with countably infinite number of tables, each with countably infinite number of seats. Customers enter the system one at a time and select a table to sit. If z_i is the index of the table chosen by the ith customer, then PYCRP defines the following distribution.

$$P(Z_i = k|z_{-i}) = \begin{cases} \frac{n_k^- - a}{i-1+b} & 1 \leq k \leq K^- \\ \frac{K^- a + b}{i-1+b} & k = K^- + 1 \end{cases}$$

where z_{-i} is the seating arrangement of the previous i - 1 customers, n_k^- is the number of customers in z_{-i} who are seated in table k and K^- is the total number of tables in z_{-i}. First customer sits at first table with probability 1 (i.e $z_1 = 1$). The joint probability of the sequence of integers produced by PYCRP is given by,

$$P(z) = \frac{\Gamma(1+b)}{\Gamma(n+b} \left(\prod_{k=1}^{K-1} (ka + b) \right) \left(\prod_{k=1}^{K} \frac{\Gamma(n_k^- - a)}{1-a} \right)$$

where K is the total number of tables in Z and Γ is the gamma function.

Consider a PYCRP that generates initial trees with some non-terminal c as its root. Each table i in the system is labeled with an initial tree with $l(z) = l_1 l_2 ... l_k$. Whenever a new customer opens up a new table, label for that table is chosen from the base distribution $P_{E_{in}}$ which is conditioned on c. Let e_i be the label of ith customer, i.e., $e_i = l_{z_i}$. Then,

$$P(e_i = e|c, z_{-i}, l(z_{-i})) = \frac{n_e^- - K_e^- a_c + (K_c^- a_c + b_c) P_E(e|c)}{n_c^- + b_c}$$

where K_c^- is the total number of tables for non-terminal c, n_e^- is the number of times e has been used to rewrite c and n_c^- is the total count of the rules rewriting c. Note that the same discussion is applicable for the PYCRPs that generate auxiliary trees too. Given a fixed distribution $P_c(\alpha|c')$ (probablity of rule $c' \rightarrow \alpha$, where c' is some non-terminal) of the CFG rules, the base distributions for initial trees is,

$$P_{E_{in}}(e|c) = \prod_{i \in I(e)} (1 - s_{c_i}) \prod_{f \in F(e)} s_{c_f} \prod_{c' \rightarrow \alpha \in e} P_c(\alpha|c')$$

where $I(e)$ is the set of internal nodes in e excluding the root, $F(e)$ is the set of frontier non-terminal nodes and s_c is the probability of stopping the expansion at a node .

3. Experimetal Results

We conducted experiments on Wall Street Journal(WSJ) corpus of the Penn Treebank . We used sections 2-21 for larger training set 23 for testing and 2 as smaller training set. We ran the sampler for 10,000 iterations. We initially trained the model with the smaller training test and tested the model with the test data. Then we trained the model with the larger training set. The results are compared with parsers based on MAP PCFG and TSG (Cohn et al., 2010) . The F1 values are reported in Table 1. It clearly shows that this model performs better than (Cohn et al., 2010).

Models	Small Training Set	Large training set
PCFG	60.3	63.5
TSG	74.6	84.4
TAG	79.4	89.9

Table 1. F1 measure for the parsers (both using small training set and large training set)

4. Conclusion

In this paper, we have proposed a Bayesian non-parametric model to induce Probabilistic Tree Adjoining Grammars from the collection of parse trees. Experimental results shows that parsers that use the induced PTAGs perform better than state of the art parsers that uses PCFGs and TSGs.

References

Cohn, T., Blunsom, Phil, and Goldwater, Sharon. Inducing tree-substitution grammars. *The Journal of Machine Learning Research*, 11:3053–3096, 2010.

Joshi, A K. and Schabes, Yves. Tree-adjoining grammars. *Handbook of formal languages, vol. 3: beyond words*, 3:69–123, 1997.