Assignment 3 - PCFG Parser

Prasanna Patil
CSA, IISc,
Banaglore
patilk@iisc.ac.in

Abstract

This document contains the final report of assignment e. The task was to train a PCFG model using Maximum Likelihood estimation of production probabilities from Penn TreeBank dataset.

1 Task 1

In this task, I implemented PCFG MLE and CKY parser for PCFG. I also experimented with two smoothing techniques as described below. The experimental setup was as below.

1.1 Experiment Setup

1.1.1 Dataset preprocessing

- The experiments were performed by splitting penn treebank dataset into 90% training set and 10 % validation set.
- Only top 10k words were considered for building vocabulary and grammar productions.

1.1.2 Evaluation Metric

PARSEVAL metric was proposed by (Black et al., 1991). This metric evaluates generated tree based on generated constituents in the tree. That is for each subtree of parse tree, we check leaves and generate a constituent set from leaves of each subtree and evaluate precision, recall over this constituent set of gold tree (correct reference tree) and generated tree.

Labeled Dependencies, this metric simply generated productions from gold tree and generated tree and calculates precision and recall over this productions set.

1.2 Smoothing Methods

I tried following smoothing methods.

1.2.1 Backoff to Unknown

This method simply, backs of to a special UNK token if the token from sentence isn't present in the vocabulary. As mentioned in section 1.3, I have considered only top 10k words from dataset. Hence, remaining words are automatically pushed to UNK token while generating grammar. This UNK rule is then used during parsing to generate rules for any token which doesn't appear in vocabulary. Henceforth this method will be identified as UNK method.

1.2.2 Equal Importance

This method gives equal importance to all those rules which are likely to occur. The set of rules which can terminate to given token are generated from the PCFG grammar itself by extracting any nonterminal which has terminated to a token at least once in the dataset. All these nonterminals can then be extended to any particular token in sentence which doesn't appear in vocabulary. Henceforth, this method will be identified as EQL method.

1.3 Evaluation

Following are the results obtained on validation dataset by generating PCFG and smoothing as mentioned above on training dataset.

Table 1: PARSEVAL performance

Smoothing	UNK	EQL
RECALL	0.697	0.692
PRECISION	0.699	0.689
F1-Score	0.698	0.691

As it can be seen, both smoothing methods perform almost equivalent but UNK backoff method has slight more advantage. Hence, I have

Table 2: Labeled F score performance

Smoothing	UNK	EQL
RECALL	0.634	0.632
PRECISION	0.636	0.627
F1-Score	0.635	0.629

used UNK backoff smoothing method as default smoothing method.

2 Task 2

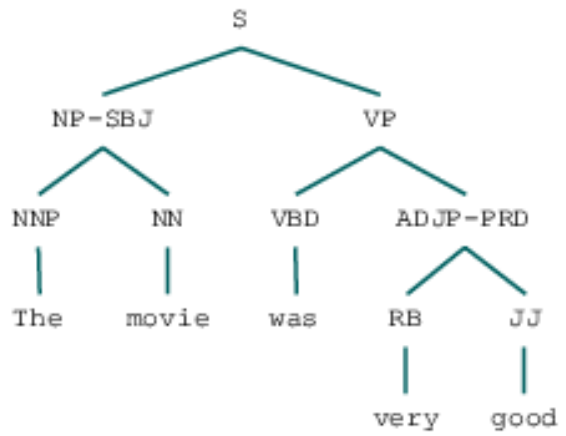
I found out following sentences that had slight difference in the parse tree generated by above parser and the one provided by Stanford ((Klein and Manning, 2003)) as web interface [1].

2.1 Short sentences

I parsed following short sentences:

- The movie was very good
- Joen was alone at home

Their parser trees generated by PCFG parser and stanford parser are as below.



(a) Figure 1

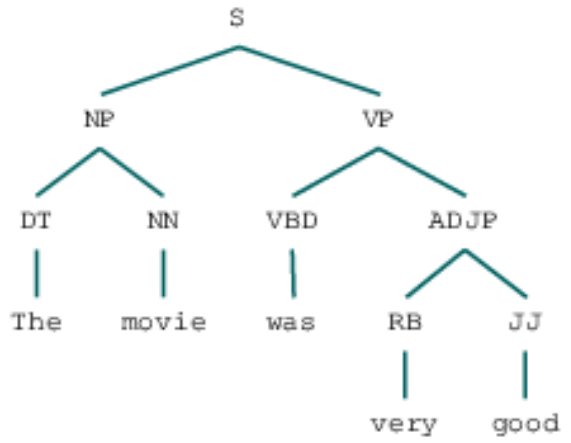
Figure 1: Figure: Parse tree by PCFG parser for 1st sentence

2.2 Long Sentences

I used following long sentences:

- Experiments done at Large Hadron Collider serve us better understanding of universe and physics.

[1] http://nlp.stanford.edu:8080/parser/index.jsp



(a) Figure 1

Figure 2: Figure: Parse tree by Stanford parser for 1st sentence

- The kind of research done at IISc puts very high emphasis on publishing papers in top journals.

There parse trees are as below.

3 Conclusion

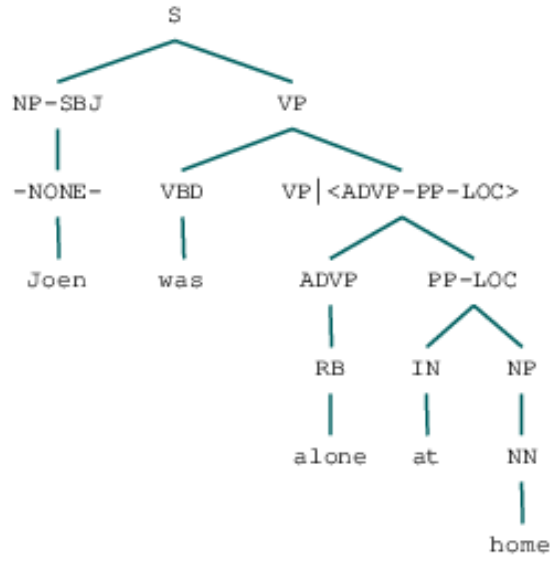
- As it can be seen from above results, PCFG parser makes mistakes by placing wrong constituents and attachment. In long sentences words are attached incorrectly. Hence, it is unable to identify constituents.
- In short sentences, it identifies constituents and attachment correctly but placing wrong non terminals in the parse tree.

4 References

References

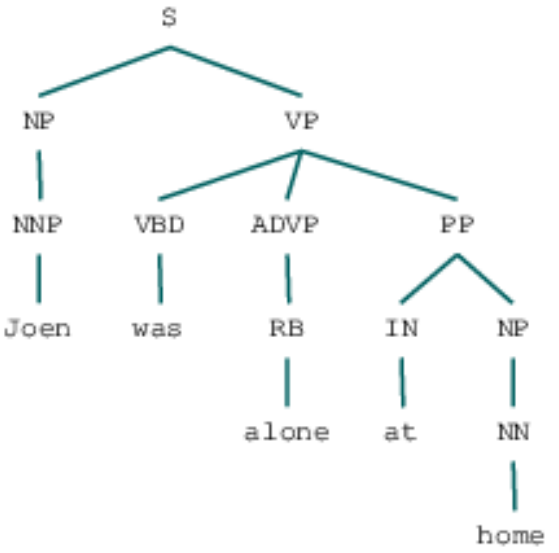
E. Black, S. Abney, D. Flickenger, C. Gdaniec, R. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. A procedure for quantitatively comparing the syntactic coverage of english grammars. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991*.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA. Association for Computational Linguistics.



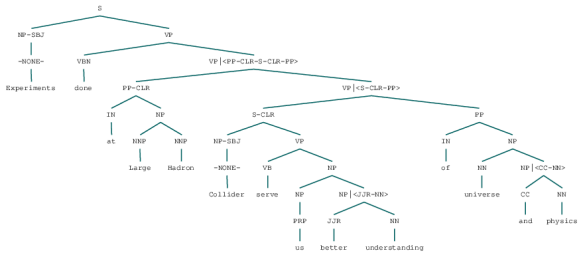
(a) Figure 1

Figure 3: Figure: Parse tree by PCFG parser for 2nd sentence



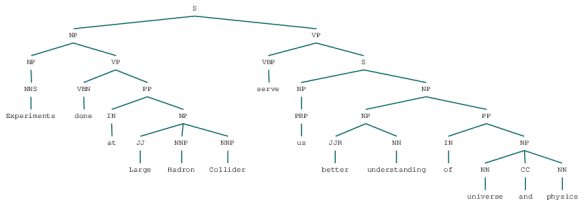
(a) Figure 1

Figure 4: Figure: Parse tree by Stanford parser for 2nd sentence



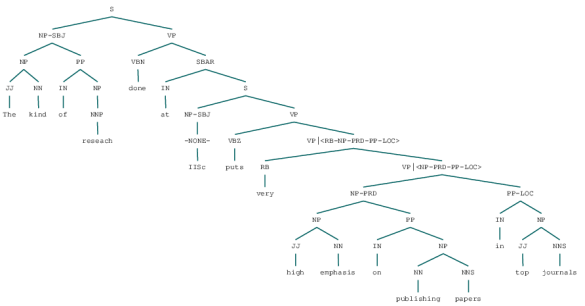
(a) Figure 1

Figure 5: Figure: Parse tree by PCFG parser for 1st sentence



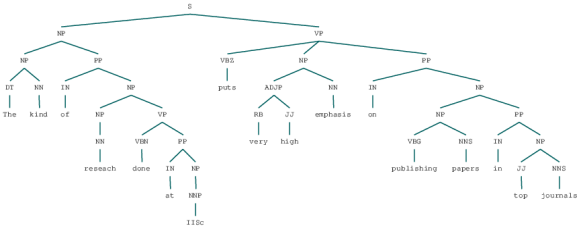
(a) Figure 1

Figure 6: Figure: Parse tree by Stanford parser for 1st sentence



(a) Figure 1

Figure 7: Figure: Parse tree by PCFG parser for 2nd sentence



(a) Figure 1

Figure 8: Figure: Parse tree by Stanford parser for 2nd sentence