Master MVA, 2017/2018

Algorithms for Speech and Natural Language Processing - TD 4

Sébastien Ohleyer

The goal of this assignment is to develop a basic probabilistic parser for French based on the CYK (Cocke–Younger–Kasami or CKY) algorithm and the Probabilistic Context-Free Grammar (PCFG) model.

1 Parser

The probabilistic parser proceed in several successive steps. We need to transform each line of the corpus in trees, binarize them, train a PCFG and run CKY on a test sentence. Note that for this work, we use the following references [JM00; Col] in addition to the course slides and we read carefully analyse the following GitHub [aet] implementation to inspire us for ours.

Trees. This preprocessing step parse the string sentence into a tree. For this step, we coded a Tree class (see `tree.py`) able to represent trees, run through them and binarize them. Indeed, to run the CKY algorithm, we need to have trees in Chomsky Normal Form (CNF). Note that any CFG can be convert to CNF.

PCFG. We split our dataset into two parts of 90% for the training and 10% to evaluate performance. Note that we did not see any need for a 10% set for development purposes. From training set transformed into CNF, we train the PCFG. This is done by a PCFG class (see `pcfg.py`). For this step, we naively count the number of occurence of each rules and divide by the number of rules for each rule sources.

CKY algorithm. With the learned PCFG model, we can now use the CKY algorithm to predict the rules on a test sentence. We implement this algorithm as a method in the PCFG class. For the test set, we remove every parathesis and labels of the sentences, to keep only the sequences of words. Then, we run our algorithm on these sequences.

2 Analysis and improvements

To evaluate our parser, we used the PARSEVAL metric [Bla+91]. We did not have time to run on every example of the test set, because of a lack of time (CKY is very long to run and it was quite difficult to achieve this project in one week).

On a few runs, qualitative results seem to be pretty satisfying. For example :

Ceux d' Ancerville ont participé à l' événement en faisant découvrir leur savoir-faire et leurs recettes .

Gives:

((SENT (NP (NC Ceux) (NP (DET d') (NPP Ancerville))) (VN (V ont) (VPP participé)) (PP (P à) (NP (DET l') (NC événement) (PP (P en) (VPpart (VN (VINF faisant) (VINF découvrir)) (NP (DET leur) (NC savoir-faire))))) (COORD (CC et) (NP (DET leurs) (NC recettes)))) (PONCT .)))

On this sentence, we do not see any error. However, when we compute the PARSEVAL metric between the computed parse tree and the golden standard it gives 0.8817, which is statisfying but not perfect. The parsing is not exactly the same.

On another example like:

Espoir

It gives:

((SENT Espoir))

whereas the golden standard is:

((SENT (NP (NC Espoir))))

Here, the PARSEVAL metric is 0.6667. We can see that for a very simple example, the CKY algorithm does not catch the structure, maybe because of the lack of context. It can be a way of improvement.

On other example, results can be impressive with a PARSEVAL score around 0.96.

Important note. As I have never worked on trees and on parsing before, I was helped by Tristan Sterin for this homework, and we work closely together on it. You may find correspondences on the code because we developed it together, however the report was individual. For any question, do not hesitate to contact me.

References

[aet] aetilley. *A Python pcfg object that consumes tree-banks in order to be trained and parses raw text into the same tree format.* https://github.com/aetilley/pcfg.

[Bla+91] Ezra Black et al. "A procedure for quantitatively comparing the syntactic coverage of English grammars". In: (1991).

[Col] Michael Collins. *Probabilistic Context-Free Grammars (PCFGs).* http://www.cs.columbia.edu/~mcollins/courses/nlp2011/notes/pcfgs.pdf.

[JM00] Daniel Jurafsky and James H Martin. "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition". In: (2000).