Question Identification Using a Probabilistic Context Free Grammar

Andrew Hanson

University of Wisconsin Madison
CS 769 Natural Language Processing
hanson@cs.wisc.edu

Abstract

This paper shows that using the tree structure generated from a Probabilistic Context Free Grammar parser adds meaningful information to language processing tasks, in particular, question identification. By using a part-of-speech representation of a sentence as a base line, this paper's results show that adding features derived from the tree output of a Probabilistic Context Free Grammar parser improves the classification of question vs. non-question sentences.

Motivation

This project was an attempt to bring a linguistical background to the task of natural language processing. In particular it uses a Probabilistic Context Free Grammar (PCFG) which is very similar to syntax, a sub-field of linguistics. A PCFG model can be used to generate a tree structure, similar to those found in syntax, for a sentence.

Generating Tree Structures

Data Source

The data for this paper was gathered from online Frequently Asked Questions pages. In particular, data was gathered from http://www.parallelkingdom.com/faq.shtml and http://www.copyright.gov/help/faq/. The data was then trimmed down to contain a sentence from each question and each answer. This process provided an equal number of labeled questions and answers which talk about similar subjects. In total 200 data points, 100 question-answer pairs, were gathered.

Preprocessing and Part-of-Speech Tagging

The data was tokenized using the Penn Treebank tokenizer. Note that this tokenizer uses question marks to mark the end of sentences but otherwise does not treat them specially. This means that question marks were not used to help identify questions. The data was then stemmed using the Porter algorithm. The stemmed data was then feed into a Maximum Entropy Part of Speech Tagger (MXPOST) described in the paper by Ratnaparkhi. The model used to do the part-of-speech (POS) tagging was the model provided with the MXPOST tool. The stream of tags, excluding the original words, is the final output of the preprocessing step.

Probabilistic Context Free Grammar

The tag streams for each sentence were feed into the Stochastic (Probabilistic) Context Free Grammar training program provided with the Edinburgh Speech Tools Library. This tool implements a forward-backward algorithm to learn the rules of the grammar. Having only a small amount of data to train a grammar on was a concern. Therefore ten different models, each with different random initializations, were trained on the data. The top grammar, ranked by cross-entropy, was used to generate the tree structures used in the rest of this paper. Figure 1 shows an example of the input sentence, intermediate part-of-speech tags, and resulting tree structure.

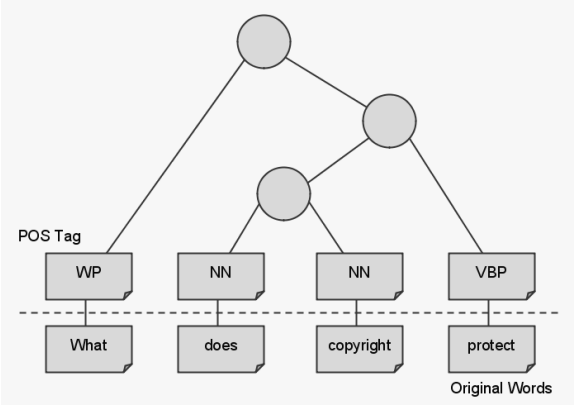

Figure 1: An original sentence, the part-of-speech tags generated, and the tree structure generated by the Probabilistic Context Free Grammar parser.

Feature Extraction and Classification

Features

Finding good features was tricky. In general the features used fall into three categories: tag only - those which use information just from the part of speech tags, tree only – those which use only the structure of the tree, and hybrid – those which used both tree and tag information. Since the motivation behind this paper was to demonstrate the benefit generating a tree structure, tag only features were used as the base line for comparison.

One tag only feature was the presence of question words. More precisely this feature was the presence of part-of-speech tags that corresponded to question words. Another feature was generated by looking at the part-of-speech of the first word in the sentence. This created an N length feature vector of 1's or 0's, where N is the number of different part-of-speech tags. This was extended to include the first four words of each sentence by concatenating each of the N length vectors to form an N * 4 length vector. If the sentence was less than 4 words long, the trailing vectors were left as zeros.

The tree only features provided more of a challenge. Different factors such as branching bias and sub tree weights we tried.

Hybrid features included looking at the size, in terminal nodes, of the sibling of question related part of speech tags. Another was to generate a part-of-speech feature vector similar to the tag only method described above for the second level nodes in the tree, the left/right child of the left/right child of the root. If the node in question was not present or not a terminal node then the zero feature vector was used.

Classification with SVM

The classification of the resulting features was done using SVM light. Linear, polynomial, and radial basis kernels were all used. The polynomial kernel was found to be the best. However the linear kernel was as good as the polynomial kernel in most cases. For that reason the polynomial kernel was used in the final results of this paper.

	POS first 4 Words	POS of 2nd Level	POS first 4 Words + POS of 2nd Level
Avg Accuracy	95%	76%	97%
Std Accuracy	5.7%	12.9%	5.3%

Figure 2: 10 fold cross validation of the most promising features.

Results and Conclusions

Of the tag only features, the tag of the first four words had the highest success rate. This is not surprising for English since most question start with a question word.

Of the tree only features none of them came close to the performance of the tag only features. Also when concatenated with the tag only feature vector no improvement was noticed at all. On further examination of the tree structures generated by the PCFG parser, they were found to be mostly right branching trees with very similar structures. With this observation it is not surprising that features derived from tree structure alone performed poorly.

Of the hybrid features the most successful was POS of the second level nodes. However this alone still did not outperform the tag only features. But when combined with the POS vector of the first 4 words, classification was improved. This change is shown in figure 2 and is statistically significant using a student-t test at the 0.05 level.

Although the problem solved by these techniques is somewhat artificial, simply looking for question marks on the data set would have worked even better, the results do suggest that syntactic structure behind sentences can be useful in natural language processing.

References

Adwait Ratnaparkhi. A Maximum Entropy Part-Of-Speech Tagger. In Proceedings of the Empirical Methods in Natural Language Processing Conference, May 17-18, 1996. University of Pennsylvania

Manning & Schutze, Foundations of Statistical Natural Language Processing, the MIT press, 1999.

Thorsten Joachims. SVM Light. http://svmlight.joachims.org/

University of Edinburgh. MXPOST. http://www.inf.ed.ac.uk/resources/nlp/local_doc/MXPOST.html