Homework 3: Active Learning for Statistical Parsing

Ankit Goyal
ankit@cs.utexas.edu
Natural Language Processing

April 8, 2015

1 Introduction

In this experiment we analyze the performance Active learning methods. For statistical parsing the training instance is sentence and the annotation is the parse tree supplied by a linguist expert. Many learning problems require supervised training however, choosing the appropriate training data may be difficult to obtain. Active learning methods uses uncertainty-based evaluation functions to estimate the training utility values of a sentence. This selection of training data reduces the size of corpus needed to train without degrading the quality of the induced grammars.

In this homework, we evaluate four different methods to measure uncertainty:

- **Random Selection**: In this evaluation method, in each iteration we simply choose random sentences (with total words >=1500) in each iteration. This doesn't necessarily use any active learning method.
- **Sentence Length**: Here as a naive method, we use sentence length as the criteria to measure uncertainty. This function approximates the length with uncertainty. It assumes that longer the sentence, the more uncertain it's parse tree is (which may not always be true).
- **Selection Probability**: use the probability assigned to the selected parse tree, the higher the probability, the lower is uncertainty. We choose more uncertain sentences first to better train our parser. we normalize the probabilities by taking $(n-1)th$ root where n is the length of the sentence, since otherwise the long sentences will always be the most uncertain ones.
- **Tree Entropy:** as defined by Hwa(2000), we consider top k parses of the sentence as a measure of uncertainty. We use the pseudocode in codeblock 1 to calculate the tree entropy.

2 Experiments:

- Section 00 of WSJ corpus was used as a initial data and section 01-03 were used as training data. Section 20 was used for testing.
- Four selection functions were implemented (random, sentence length, selection probability and tree entropy).
- F1 score is used to compare different selection functions.

```
// sum the probability of top 10 parse trees
for (Tree tree : topKparses)
        sum += Math.exp(tree.score());

// normalize the probability of each parse tree.
for (Tree tree: topKparses)
        normalized_scores_list.add(Math.exp(tree.score())/sum)

// calculate entropy as described in Hwa(2000),
// as expected negative log likelihood of each parse probability.
double entropy = 0.0;
for (double score : normalized_scores_list)
        double ent = -1 * Math.log(score)/Math.log(2) * score
        entropy += ent;

// normalize the entropy with the sentence length.
normalized_entropy = entropy/sentence_length.
```

Codeblock 1: Entropy Calculation

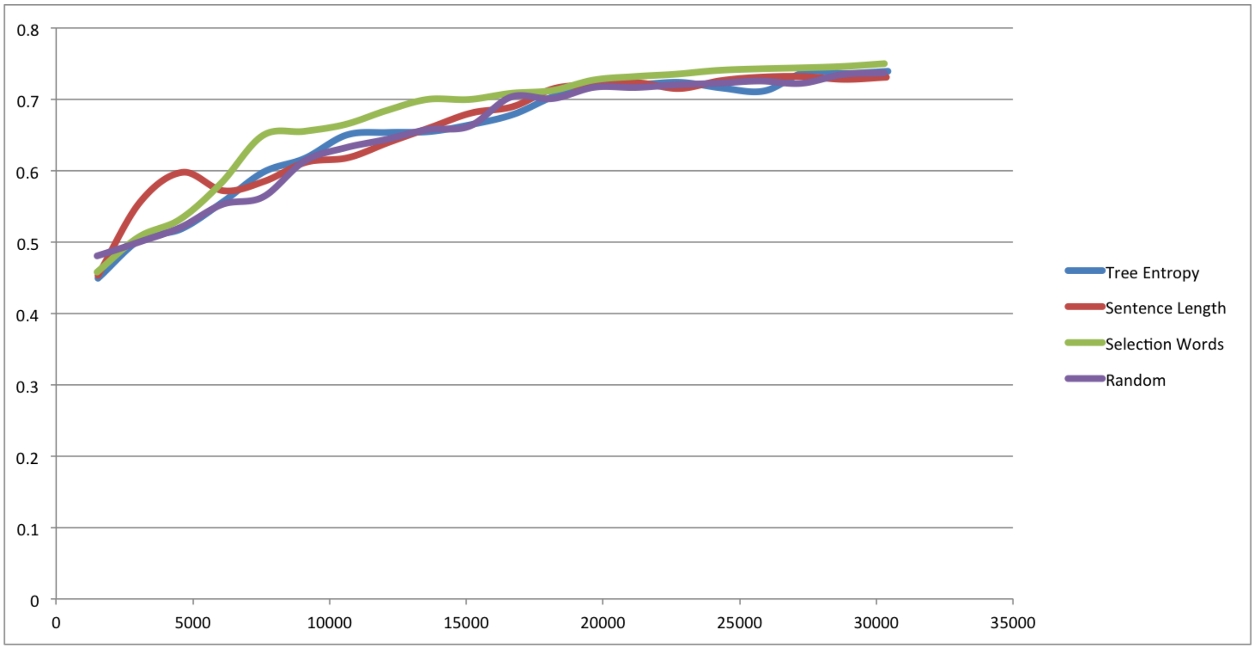

Figure 1: F1 vs Number of training words

3 Observations:

1. Active learning methods does seem to perform better than random selection since we are training on more complex data by actively selecting to train our parser better.
2. Tree entropy doesn't perform as well as expected or as it is mentioned in the paper. One of the reasons for this could be the dataset.
3. Selective probability performs the best and random performs the worst (very close to selection by length).

The difference is performance of tree entropy and sentence length or random selection is not as huge as mentioned in the paper.

4 References:

1. Rebecca Hwa. Sample Selection for Statistical Grammar Induction.