Cloud Computing Project Proposal
Language Modeling

Denis Filimonov
University of Maryland
den@cs.umd.edu

Mary P. Harper
University of Maryland
Purdue University
harper@purdue.edu

1 Problem

Latent variables assigned to tags allow the Berkeley parser (Petrov et al., 2006) archive state-of-the-art performance in parsing using a very simple PCFG model. We can extract latent preterminal (POS) tags (simply "latent tags" later) and use them to create a better language model.

Consider the following equation describing a trigram language model.

$$
\begin{aligned}
Pr(w_1^n) &= \sum_{c_1^n} Pr(w_1^n c_1^n) \\
&= \sum_{c_1^n} \prod_{i=1}^n Pr(w_i|w_1^{i-1}c_1^i)Pr(c_i|w_1^{i-1}c_1^{i-1}) \\
&\approx \sum_{c_1^n} \prod_{i=1}^n Pr(w_i|w_{i-2}^{i-1}c_{i-2}^i)Pr(c_i|w_{i-2}^{i-1}c_{i-2}^{i-1})
\end{aligned}
$$

Since the sequence of words is fixed in a typical LM task, the space and time complexity is $|C|^3$ per word using the trigram model, where C is the set of classes. With latent tags the set becomes too large for straightforward implementation.

The idea is to implement hierarchical clustering of the context space $(w_{i-2}^{i-1}c_{i-2}^{i-1})$ similar to (Brown et al., 1992) and (Heeman, 1998). However, learning the best partitioning is computationally hard. With the Hadoop cluster we will be able to use more data for training and get results quicker, which is very important during the development cycle.

2 Resources

We are going to train the model on the annotations produced by the Berkeley parser.

3 Mapreduce Perspective

The problem fits the Mapreduce framework very nicely. Mappers will compute scores (entropy) of possible partitionings and reducers will pick the best one.

3.1 Interesting Extensions

Large computational resources will enable us to explore ideas no-one tried before. For example, (Heeman, 1998) use binary trees to partition the space, they do not attempt to merge branches with similar distributions (which would alleviate the sparsity) because this was computationally prohibitive.

References

Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

P. Heeman. 1998. Pos tagging versus classes in language modeling. Technical report.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440, Sydney, Australia, July. Association for Computational Linguistics.