CMPT 825: Natural Language Processing **Spring 2008**

Lecture 9 — Jan 25, 2008

Lecturer: Anoop Sarkar *Scribe: Anton Venema*

The paper under consideration for this scribing report is *Tagging English Text with a Probabilistic Model* by Bernard Merialdo. He describes two methods for measuring the quality of a tagging procedure:

1. At the sentence level, the percentage correctly tagged, evaluated with Viterbi tagging
2. At the word level, the percentage correctly tagged, evaluated with Maximum Likelihood (ML) tagging

It's useful to note that sentence-level performance will always be lower than word-level, since the former is dependent on the latter. Merialdo goes on to describe what he calls a triclass model, commonly known as a trigram model:

$$p(W,T) = \prod_{i=1}^{n} p(w_i|t_i) \cdot p(t_i|t_{i-2}t_{i-1})$$

Two training methods are described. The first method is relative frequency (RF) training, which uses tagged text to count observations and generate frequencies for tag sequences and word/tag pairs (supervised).

$$p(W,T) = \prod_{i=1}^{n} \frac{N(w_i,t_i)}{N(t_i)} \cdot \frac{N(t_{i-2},t_{i-1},t_i)}{N(t_{i-2},t_{i-1})}$$

Deleted interpolation (Jelinek-Mercer, 1980) is then applied to smooth for unseen data. ($|V(t_i)|$ is the number of words with tag t_i.)

$$p(W,T) = \prod_{i=1}^{n} (\frac{\lambda \cdot N(w_i,t_i)}{N(t_i)} + \frac{1-\lambda}{|V(t_i)|}) \cdot (\frac{\lambda \cdot N(t_{i-2},t_{i-1},t_i)}{N(t_{i-2},t_{i-1})} + \frac{1-\lambda}{N_T})$$

The second method is maximum likelihood (ML) training, which does not require tagged text (unsupervised). Merialdo used the Forward-Backward

algorithm to maximize the probability of the training text, using the same training data as the relative frequency training, but without looking at the tags. An RF-trained model was used to initialize the probabilities.

The results indicated that RF training was quite accurate (95.4%) with about 2000 training sentences. Using 100 times as much training data yielded only a 1.6% improvement. ML training improved the RF-trained models when very few (between 0 and 5000) training sentences were used for the initial model. It showed its most significant improvements with 0 to 100 sentences used initially. After 3 iterations, it degraded all performance except in the 0-sentence initial-model case.

Merialdo briefly mentioned the idea of constraining ML training to hinder accuracy degradation. The *tw-constraint* keeps the probability of a given word/tag pair constant if it occurs frequently (i.e. in the top 1000 words). The *t-constraint* keeps the probability of a tag constant. Unfortunately, details on the implementation of this method were not described, other than to mention its complexity and adverse effect on the overall running time of the algorithms.

The paper was a good comparison of RF versus ML training. The take-away message is that there is no data like well-tagged data, so use it as much as possible. If using ML training, test it on a data set after each iteration to ensure you are not making your model worse.