CMPT 825: Natural Language Processing **Spring 2006**

Lecture 10 — Feb 15, 2006

Lecturer: Anoop Sarkar *Scribe: Fereydou Hormozdiari*

In this weeks lecture we went through Language model and different Smoothing techniques. However, before starting to review smoothing algorithms we need to see some of the elementary probability rules.

$P(a, b, c|d) = P(a|b, c, d) \times P(b|c, d) \times P(c|d)$

The above formula is know as chain rule. However, the right hind side will be too complicated if we want to calculate for the real language modeling perposes. Hence, we will do an approximation.

$P(a|b, c, d) \approx P(a|b, c)$ less bias

$P(a|b, c, d) \approx P(a|b)$

$P(a|b, c, d) \approx P(a)$ more variance

Now for realizing the need for Language Models lets look at one very simple example. The "Simple error Spelling Correction" problem. However, trying to solve this problem without looking at the context is almost impossible. Thus, we need to be able to calculate the probability of each sentance $P(w_1, w_2, w_3, ..., w_n)$, which using the chain rule explained above is equal to $P(w_1)/timesP(w_2|w_1)/timesP(w_3|w_2, w_1)...P(w_n|w_1, ..., w_{n-1})$. Now the question is how we can train the above probabilities regarding the curse of dimensionality(Sparse Data).

For simplifying the above equation, we can consider different assumptions. One of these assumptions is Markov Assumption. Which, states that each word is only dependent to two of its previous words. Now, considering the Markov Assumption, how many distinct probabilities is need? (size of Vocabulary is $|V|$.) We need $|V|^3$ probabilities, however, the number of observations we have is W_T(the number of word tokens we have). This means that we are going to have a very large set of zeros(Sparse Data). Hence, we need a smoothing technique, to reduce the sparse data effect.

10.1 Information Theory

Entropy is the measurement of disorder.

$H(X) = -\sum_{\varepsilon} P(x)log_2P(x)$

When the log in base two is used, the computation is in bits. In another words, the above formula is calculating the number of bits needed to transmit message from space ε, where P(x) defines the probability of the event.

The cross entropy is a distance measure between P_t and P_e.

$H(P_t, P_e) = -\sum_{\varepsilon} P_t(x) log_2 P_e(x)$

There are other measurement types of based on entropy such as Relative Entropy and Mutual Information.

10.2 Smoothing

There are different methods of Smoothing which a complete survey of them is given in a paper by S. Chen and J. Goodman.

$P(w_i|w_{i-1}) = \frac{C(w_{i-1},w_i)}{C(w_{i-1})}$

The first model that we are going to consider is called Add-One Smoothing:

$P(w_i|w_{i-1}) = \frac{1+C(w_{i-1},w_i)}{V+C(w_{i-1})}$

Second method is Additive smoothing, which is similar to Add-One Smoothing method.

$P(w_i|w_{i-1}) = \frac{(\delta+C(w_{i-1},w_i))}{(\delta \times V)+C(w_{i-1})}$. $0 < \delta <= 1$

The third model we will look at is called Good-Turing Model. This model states that for any n-gram that occurs r times, we should pretend it happend r^* times $r^* = (r+1)\frac{n_{r+1}}{n_r}$, which n_r represents the number of n-grams that appear exactly r times.

The remaining models are various versions of Backoff Smoothing method. First lets look at one of the main deficiencies of add-one and Good-Turing model. In both of these models if a n-gram does not appear, then they assign a same probability to them. For instance, P(the—string) = P(Fonz—String).

One of the variations of Backoff method is, Jelinek-Mercer method. In this method if we assign P_{ML} as standard maximum likelihood, and P_{JM} as the Jelinek-Mercer likelihood, then $P_{JM}(w_i, w_{i-1}) = \lambda P_{ML}(w_i|w_{i-1}) + (1 - \lambda)P_{ML}(w_i)$.

There are other alternations of this smoothing method such as Katz Backoof, which is including Good-Turing method with Backoff smoothing method.