1 Architecture

1.1 Hidden Markov Model

First order hidden Markov model (i.e. bigram model) is used to model the sequence of tags in a sentence. Transition probability $P(\ t_i\ |t_{i-1}\ )$ and observation likelihood $P\ (\ w_i|t_i\ )$ are calculated from the training data. These sequence of states (i.e. tags) respresents the hidden part of the Markove model. Using, these 2 probabilities the objective is to find the following sequence of tags T:

$$T = \ arg\max_t \left\{ \left(\prod_{i=1}^{T} P(t_i|t_{I-1}) \cdot P(w_i|t_i) \right) \cdot \ P(</s>|t_T) \right\}$$

The Viterbi algorithm is used to find such sequence of tags.

1.2 Smoothing

The sparse data problem means not all combination of bigram sequence will be seen in the training data. Consequently when tagging new data, there will be cases where the bigrams have not been seen before. Setting such probability to zero will cause undesirable effect because the probability of the whole sequence will become zero.

1.2.1 Absolute discounting

This method works by subtracting a fixed fraction from the maximum likelihood estimate. For counts with high value, subtracting a small discount will not affect the probability value by much. For counts with low value, even though discount might shift the value by much more, it is still quite acceptable since those with low counts are normally not as important as those with high counts.

$$P_{absolute}(w_i|w_{i-1}) = \begin{cases} \dfrac{C(w_{i-1}w_i) - D}{C(w_{i-1})} & if\ C(w_{i-1}w_i) > \ 0 \\ \alpha(w_i) P_{absolute}\ (w_i) & otherwise \end{cases}$$

1.2.2 Minimum probability

A bigram sequnce that has never been observed may approximates the bigram sequence that occurs only once. Hence, for these unobserved bigram, we can assign it the same probability value as the probabiliy singly occuring bigram. Some weight may be used to ensure that the sum of probability is still 1.

1.2.3 Constant value

Another possible approach is

1.3 Handling unkown words

2 Evaluation

3 Conclusion

4 References