NLP: N-Gram Language Models

Tyler McDonnell
tyler@cs.utexas.edu

N-Gram Language Models

Language models construct a probability distribution over sequences of words and in turn provide a tool for estimating the relative likelihood of phrases in a language. These models can be used for a diverse set of tasks in NLP, such as speech recognition, machine translation, authorship attribution, or handwriting recognition. N-gram language models construct these distributions using co-occurrences of words in sequence. For example, a bigram model relies on the Markov assumption and computes the probability of a word given the word prior to it. More generally, an N-gram language model conditions on the previous N - 1 words.

Forward and Backward Bigram Models

I start with a basic Forward Bigram model that features linear interpolation with a unigram model ($\lambda_{unigram} = 0.1$, $\lambda_{bigram} = 0.9$) for smoothing and replaces the first occurrence of each word during training with <UNK> as the mechanism for handling out-of-vocabulary (OOV) words. It may seem interesting to consider a Backward Bigram Model, which models sentences from right-to-left. In fact, it is simple to implement such a model given a Forward Bigram model. I simply reverse the input sentences during training and testing. Thus, this Backward model also features the same smoothing and OOV handling mechanisms as the Forward Model.

To compare the performance of these two models, we develop an informal measure of perplexity: *word perplexity*. The standard definition for perplexity is shown below. Word perplexity is distinguished by ignoring sentence boundary predictions. Since the Forward and Backward models differ fundamentally in which sentence boundary they must predict (i.e., end-of-sentence vs. beginning-of-sentence), we consider word perplexity to be a fairer point of comparison.

$$\text{Perplexity(W)} = \sqrt[N]{\prod \frac{1}{P(w_i|w_{i-1})}}$$

Comparison of Forward and Backward Performance

Table 1 shows the perplexities for the Forward and Backward Bigram models for three different datasets: *Wall Street Journal (WSJ)*, *Brown*, and *ATIS*. The Backward Model performs slightly better in both the WSJ and Brown datasets, whereas the Forward Model performs better on the Atis dataset. One hypothesis for this is based on branching sentence structure. In the English language, right-branching sentences, in which modifiers to the subject occur to the right of the subject, are more common than left-branching sentences. Thus, the Backward Model might provide more information in general for written English. However, the relatively small ATIS dataset is composed of transcribed conversations, rather than written English, which may tend to be dominated by predictable tendencies at the beginning of spoken sentences.

Both models perform significantly better on training data than testing data. This is not surprising, since count-based N-gram language models are essentially building maximum likelihood estimations (MLE) over the training set, and are thus by definition overfit to the training data. Thus, cross-validation with an independent set of test sentences is a more useful metric.

ATIS	Forward	Backward	**WSJ**	Forward	Backward	**Brown**	Forward	Backward
train	10.592	11.636	train	88.890	86.660	train	113.360	110.783
test	24.054	27.161	test	275.118	266.352	test	310.667	299.686

Table 1: Forward and Backward Bigram Model "Word Perplexity."

Bidirectional Bigram Model

Given a Forward and Backward Bigram Model, we can also create a simple ensemble model that combines the two. For this Bidirectional Model, I simply train a separate Forward Model and Backward Model and at test time, I run each sentence through both models and linearly interpolate the predictions from both models for each word. The final probability assigned to a word by the Bidirectional Model is shown below, where λ_F and λ_B are the interpolation weights.

$$P_{Bidirectional} = \lambda_F * P_F + \lambda_B * P_B$$

Comparison of Bidirectional Performance

Figure 1 shows the word perplexities for the Bidirectional Bigram Model for $\lambda_F = \lambda_B = 0.5$. It performs significantly better than either the Forward or Backward Model in all cases. Perhaps this is not surprising, since this Bidirectional Model effectively conditions the prediction for a word on both the preceding and proceeding word. In other words, this bidirectional model is a kind of trigam model! In fact, this model might provide even more resiliency to small training sets in comparison to a traditional trigram model, since the space of grams accounted for by the interpolated models has only doubled, rather than grown exponentially.

Figure 2 shows the effect of varying λ_F and λ_B. Intuitively, performance hits a peak with a slight bias towards the Backward Model, and as you increase the weight of one model, the performance tends toward that of the performance of that model in isolation.

Not surprisingly, the bidirectional model also performs much better on training data. Here, we are linearly interpolating two MLEs for the training data, so we would expect our model to fit the training data very well. Again, this illustrates the importance of cross-validation.

ATIS	
train	7.235
test	12.700
WSJ	
train	46.514
test	126.113
Brown	
train	61.469
test	167.487

Figure 1: Bidirectional "Word Perplexity."

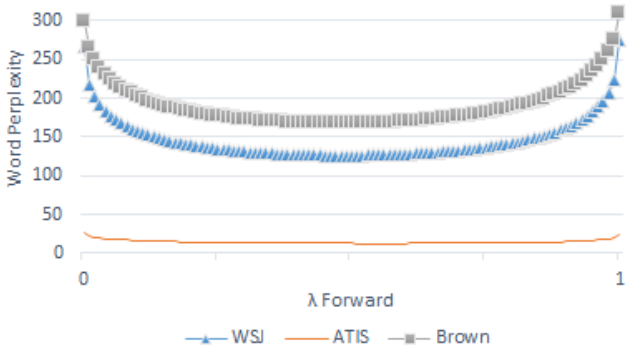

Figure 2: Interpolation Comparison