Category-based N-gram Language Models

Peter Yung

CSCE 489-508

November 30, 2017

Introduction

Main Questions:

> Can N-gram performance be enhanced by incorporating pre-determined categorical information about a document?
>
> If such categorical information is calculated on-the-fly, does this offer a significant improvement?

Possible Applications:

> Tone-of-voice analysis to improve speech recognition
>
> Sentiment analysis to improve spelling correction

N-grams

An **n-gram model** calculates the probability of a word given **n-1** preceding words*

$$P(w_i | w_{i-n+1}, \ldots, w_{i-1})$$

Perplexity is a method of evaluating n-gram models, and is calculated using the following formula for a sentence W:

$$PP(W) = \prod_{i=1}^{|W|} P(w_i | w_{i-n+1}, \ldots, w_{i-1})^{-\frac{1}{|W|}}$$

A good language model is one which has low perplexity on a set of test sentences

*In this project, Kneser-Ney smoothing was used to estimate this probability

Methods

Three different n-gram models

1. **Standard**
   - Probability calculated based on n-gram counts from entire corpus
2. **Known-class (KC)**
   - Probability calculated based on n-gram counts in one category
   - Category supplied to probability function
3. **Progressive-class (PC)**
   - Probability calculated based on n-gram counts in one category
   - Category calculated by Naive Bayes based on sentence as seen so far

Methods

- Each model was trained on a large subset of sentences from the Brown corpus

- Models were compared side-by-side with a random selection of sentences from a non-intersecting subset of the Brown corpus

- The main metric for measurement is perplexity ratio
    - Recall better language models yield a lower perplexity
    - Improvement ratio calculated as $\mathbf{PP_{model1}/PP_{model2}}$

Results: Standard vs Known-class (Trigrams)

While one might expect the Known-class model to offer a significant improvement, in reality, this is not the case.

- KC model offered an improvement on just **35% of sentences**
- However, in cases where KC did improve, there was often a large improvement. For instance, in this sentence from the category “Adventure”, the perplexity of the standard model was **97 times** the KC model:

  “How’s Sally like rubbin’ agin that thar little ticklebrush ye’re raising”

- Intuitively, and upon further inspection, examples with words and n-grams unique to their category show an improvement with KC
- Similarly, KC has higher (worse) perplexity on examples with generic words and n-grams

Results: Known-class vs Progressive-class (Trigrams)

One would expect that the PC model (Naive Bayes) would perform worse if the category was incorrectly estimated. However, while the classifier was incorrect for **70 percent** of sentences, PC performed better than KC on a majority of sentences.

Why?

- Naive Bayes chooses the most likely category based on word content
- The most likely category does not always coincide with the actual category
- Thus, word category is *not necessarily* a good way to predict the next word

Results: Known-class vs Progressive-class (Trigrams)

Example output

“What does he have in mind to do when he graduates”

Naive Bayes category estimate: “**belles_lettres**”; Actual category: “**lore**”

PC Perplexity: **114.13**

KC Perplexity: **1954.87**

Limitations

Some limitations of this study:

1. A larger training set should be used. Looking into the data, the corpus is largely made up of unique N-grams
   - Trigrams: 95.6% unique
   - Bigrams: 84.5% unique
2. Possible precision errors; Long, somewhat distinctive sentences cause the probability to trend toward 0

Conclusion & Future Work

These results show that incorporating categorical information does not, in general, improve the performance.

However, the results from the progressive-class model hint at a different approach: Instead of using pre-determined categories for each sentence, it may be more useful to first cluster sentences based on n-gram content.

- These clusters could then be used to inform the progressive-class model