Assignment 3 Report

Marc-Andre Haley • Chu Duc Thang

How did you select the final sets of parameters (n)? What are the values? Which model performed the best? Discuss the relative performance of the smoothing variants and n-gram settings:

In general we selected the values of the final n parameters by trial and error.

For the unsmoothed model, the only n value that worked was n=1. This is because at n>1 we get zero-probability bigrams, trigrams etc. At n=1, unigrams, we get around zero-probability by replacing unknown characters with “<UNK>” token.

For the laplace model, n=1 gave us many misclassifications and high perplexity. n=2 gave us the lowest perplexity out of any n value, but it resulted in 1 more misclassification than n=3. n=3 only misclassified one language, ‘deu_1901’ with ‘deu_1996’. The training data for both of these languages is the exact same except for one symbol, so the misclassification makes sense. Using n>3 gave us similar results but with higher perplexity so we used n=3.

For the linear Interpolation model, using n=1 gave us very similar results as the unsmoothed model because in this case they behave the same way. For n=2 and n=3 the model gave us very similar results as the Laplace smoothing model but with one more misclassification. This extra error probably comes from the fact that we are using less training data during interpolation since we are using some of the data to train the deleted interpolation weights. Using n>3 did not improve accuracy so used n=3 again.

The main difference between the interpolation and Laplace model is that when n grows larger (n=5+), the Laplace model gets worse, whereas the interpolation model still gives the same results, just higher perplexity. This is because when n gets larger, the Laplace model doesn’t have many ngrams from training that appear in the test data. In interpolation, when an n-gram doesn’t exist we go to the next n-1-gram so the results don’t change.

In this assignment, using these optimal n, the Laplace model gave us slightly better accuracy than the Interpolation model so it performed the best. Although the unsmoothed model still gave good results, it performed the worse out of the three with 4 misclassifications and the worse perplexity numbers.

Extension

- *How might character set information or encoding support this task or eliminate the need to develop models?*

  If we had a character set for every language and large datasets, then we could look for characters that make a language unique and once they are found we can label it as that language. Almost every language has a unique character set so this could work. If two languages have the same character set then we would need to rely on models to distinguish those languages.

- *How might transforming something from its original character set into the Roman alphabet support or hinder the task? How would different approaches to this transformation or inconsistencies in the transformation affect your models?*

  Depending on the language this could hinder or support the task. It could hinder it by making many languages too similar, therefore making it harder for models to distinguish between the languages. However, in the case of languages with very complex character sets (ex. Mandarin Chinese, Japanese) it could support the task by supporting language models made for more conventional/western languages. Small discrepancies in the character set transformations could have large effects on the performance of the language models. For example, perhaps a small difference in a Japanese symbol means a completely different word so transforming it wrong can cause the model to be incorrect.