CMPUT 501 Assignment 2

Shehroze Khan and **Afia Anjum**
Department of Computing Science
University of Alberta

1 Method & Implementation

The first step in our approach is to clean up the given input text files. We remove any extra white-spaces, newlines and tabs with a single white-space. We proceed to replace this single white-space between our words with an underscore sign, beside adding one more underscore signs at the beginning and the end of our text file. An example of our cleaned up text (in English) is as follows:
_a_quick_brown_fox_jumped_over_the_lazy_dog_
Both the training and test text files are modified in the same manner to ensure a standardized analysis.
In the training step, we proceed to generate both the n and n-1 grams of the training text files over characters. The calculation of both is performed in order to aid in the MLE calculations. With n=3 the a text above would look like:
_a_, a_q, _qu, qui, uic, ick,... and so on.

Putting underscores while tokenizing words from the files and then including these underscores when making character level n grams out of it, can provide us some useful information. For example, a trigram like _a_ can help us to determine the frequency of observing a character 'a' as the last letter of a word. This specific piece of information might have significance in languages(like English), where words are separated by spaces. So we decided to keep it in our model. However, in some languages, where words are not separated by spaces,this particular information would not be significant anyway.

We solve the problem of having a zero probability while testing for a unseen character by replacing the character by (UNK). We are doing this by selecting a threshold for the probability values so that everything below that threshold gets set to (UNK). For each of the model, we are setting this threshold by the n grams that occurs with the least frequency.

For an unsmooth model, we estimate the probability of a character level n gram by dividing the observed frequency of that n gram by the observed frequency of it's n-1 gram (as stated in equations 3.11 and 3.12 (Jurafsky and Martin, 2019)).
For the Laplace one smoothing model(as referenced in equation 3.23) (Jurafsky and Martin, 2019), we alter the aforesaid estimate to calculate the probability by adding the numerator with 1 and the denominator with V, where V is the size of the vocabulary i.e the the number of unique character level unigrams in our model.
For the deleted interpolation part, we compute n lambda values from our training corpus utilizing the deleted interpolation algorithm mentioned in section(8) and the equations referred in sections(8.26-8.29)(Jurafsky and Martin, 2019) to compute our probability.

We take logarithm of the each such probabilities and add them up.We then compute the perplexity scores from this value using the equations derived in section 3.2.1 (Jurafsky and Martin, 2019).

Under all these assumptions, we are building character level n grams and n-1 grams of each of these 55 files and using this trained model for performing our tests. We select one of the test files and assign a perplexity score with respect to all the train files. The train file which assigns the lowest perplexity score for this particular test file is kept in record. So is kept the lowest perplexity score. This record indicates that this test file belongs to that certain language in which the train file is written in.

We follow each of the aforementioned steps for each of the test files in order to get the lowest perplexity score assigned for each of them. However, some of these test files are getting misclassified by having lowest perplexities scores for a train file that represents a different language than the text file itself.

In-order to check the performance of our mod-

els, we are computing the accuracy for each of the models(unsmoothed model, Laplace one smoothed model and the Deleted Interpolation model) varying the value of n.

Finally we list all the sources and references which helped us in implementing these models.

2 Parameter Tuning

- Unsmoothed Model : If we set the value of n greater than one, even after replacing the unseen with the (UNK) token, we are simply avoiding the entire context of that particular n gram. Such a case would not maximise our performance. Hence, we use n=1 for the unsmoothed model. The accuracy is provided in the next section.

- Laplace one Smoothing Model : Looking at the accuracy values listed in the next section, we tune the value of n=7 for this model.

- Deleted Interpolation Model : Looking at the accuracy values listed in the next section, we tune the value of n=4 for this model.

3 Relative Performance of Unsmoothing and Smoothing Variants

n	Accuracy
1	87.27%

This table shows the accuracy(in percentage) of an umsmoothed model with n=1

n	Accuracy
1	65%
2	96.36%
3	94.55%
4	94.55%
5	96.36 %
6	96.36 %
7	98.18 %

This table shows the accuracy(in percentage) of Laplace one smooth model varying values of n

n	Accuracy
2	90.91%
3	94.55%
4	96.36%
5	85 %

This table shows the accuracy(in percentage) of Deleted Interpolation model varying values of n

4 Error Analysis

As a part of error analysis, we have extracted the files which are misclassified as another files. There might be multiple factors which are responsible in producing all these misclassified files. One of the reasons can be that the symbols constructing each of the languages in our given files are almost similar with some minor differences. So when we are taking character level grams, same letters or sequence of letters could appear in multiple texts.

n	File Name	Misclassified as File Name
1	udhr-bcl	udhr-cha
1	udhr-deu_1901	udhr-deu_1996
1	udhr-kqn	udhr-yao
1	udhr-nbl	udhr-xho
1	udhr-nya_chechewa	udhr-nya_chinyanja
1	udhr-sco	udhr-eng
1	udhr-xsm	udhr-hat_popular

For our Unsmooth model with n=1 & Accuracy=87.27% we find the aforementioned misclassifications

n	File Name	Misclassified as File Name
7	udhr-deu_1996	udhr-deu_1901

For Laplace one smoothing model with n=7 & Accuracy=98.18% we find the aforementioned mentioned misclassifications

n	File Name	Misclassified as File Name
4	udhr-deu_1901	udhr-deu_1996
4	udhr-zul	udhr-xho

For Deleted Interpolation model with n=4 & Accuracy=96.36% we find the aforementioned misclassifications

5 Division of Labor

Both the authors discussed and formulated the conceptual solution together. From that conceptual solution, they have implemented the unsmoothed model separately. Shehroze focused on implementing the Laplace one smoothing model while Afia worked on implementing the Deleted Interpolation smoothing model. Both the implementations were integrated and filed as one.

Following is a list of references that we used to implement our models:

References

Daniel Jurafsky and James H. Martin. 2019. Speech and language processing.

Speech and Language Processing book
Probability Smoothing for Natural Language
Vocabulary In Laplace
A Statistical Part of Speecher
Deleted Interpolation