Report

Luc Minnee, Luka de Vrij, Ewoud ter Wee, Mireille Scherpenisse

May 20, 2022

1:
We put all the results in a table. On the y-axis we have which ngram was used and on the x-axis we have which collection of testdata was used. On every collection, the trigrams beat the bigrams. Therefore, we conclude that the trigrams are better. However it is visible that the accuracy for bigrams goes up when the input is longer. It could well be that bigrams and trigrams will have the same accuracy when the input length goes up even further.

	10	30	90
Bigram	19 correct, 11 incorrect	27 correct, 3 incorrect	29 correct, 1 incorrect
Trigram	22 correct, 8 incorrect	29 correct, 1 incorrect	30 correct, 0 incorrect

2:
We tried a lot of limit values (both bigrams and trigrams the same value). These two were the closest at a limit of 20 - 50, getting really close at 21/20, yet the bigrams did not ever beat the trigrams. We think the reason behind these values is that the trigrams have a lot more original ngrams, which makes the cosine similarity function more picky. Then, theoretically, the bigrams should have won at much lower limit values, because then the cosine similarity function becomes too picky. The function will then not get any matches. Sadly, we could not prove that, since the bigrams became even less accurate, even less accurate than the trigrams.

3:
With n = 1 and n > 3 nothing gets predicted correctly. With n > 3 everything gets guessed as the language ZULU. It could be because the overwrite of the models-files was not done correctly on n > 3, which in that case the cosine-similarity would be 0. This way ZULU always gets chosen as highest score, since the list is sorted that way.

4:
Overall the group work went pretty smooth. We were able to divide the work evenly. In the lang detect script we all worked on different functions. Later on in the project when the scripts got longer and more complex it was harder to divide the work, but we still worked together on the assignment without problems.
‘

Notes:
The datafiles folder might be a bit weird, since the path to the models is datafiles/datafiles/models/.. This is because when extracted, an extra folder was created. The entire program however is designed to work with this duplicate folder. When testing, please note that any automatic testing programs may have problems with this.
Additionally, the profiles have already been created for n = 2 and 3. These are located in their own folders. To create new profiles, please use the write-profiles script to create new profiles. All old profiles will be overwritten, as long as the names of the new training files are the same as the old ones.