Reflection Milestone 3

Annemarie Wittig (3789345), Moritz L. Brunsch (3750805), Max T. Martius (3713254)

November 26, 2023

With this short essay, we want to reflect on the task of building a simple retrieval system using TIRA data for milestone 3. For us, the main part of this milestone was getting acquainted with new technologies and libraries that we had not used before. Specifically, we used the *python-terrier* library, which was new to us. We opted for it as we had a good base using the provided tutorials from our course combined with the documentation.

For this task, we first started by choosing the BM25 algorithm as a base as all of us had a good grasp of this algorithm from the lecture. We discussed on whether to use a stemmer, a lemmatizer, or neither but opted for the stemmer. For this, we first ran the tests using neither and then picked two of the pre-implemented stemmers of *python-terrier*: The *English Snowball Stemmer* and the *Porter Stemmer*, with the *Porter Stemmer* being the default that is normally used for runs. Since we had not yet implemented a proper evaluation, the choice between the three was made only by manually scrolling through the data. For that, we took a look at the stored result runs with each of the tools. To have a proper comparison, we picked the same query for each (using the query ID in column one). We then compared their ranked documents (column three) and their respective scores (column four).

Exemplary, if we take a look at the first query ID *q062210081*, it seemed to have the same top three documents. However, without using a stemmer, there seemed to be a lot more documents having a high score (e.g. there are 4 with a score of 14.x and 45 with a score of 13.x). In contrast, the other two only had three documents with a score of 14.x and 30 with a score of 13.x . This eliminated the choice of no stemmer as, to us, this proved the algorithm had more problems discriminating between the documents, making the ranking and thus the ultimate picks harder to select. However, the two other stemmers did not exhibit a substantial difference in performance, suggesting their similarity in functionality. Therefore, we decided on the *Snowball Stemmer*, guided by our intuition but no other actual facts to support this choice.

Lastly, we optimized the algorithm using the b and k_1 values that resulted from the corresponding hyperparameter tuning tutorial using the *validation test set.* Given the nature and use case for *validation data*, it made a lot more sense to use it as opposed to the results based on the *training data* alone. Comparing the tuned result to the none-tuned did yield different results, but those seemed to be smaller in difference, picking different documents from ranks 5 and 6 onwards. We still used a similar choice process to the one detailed above and then settled for the values $b = 0.8$ and $k_1 = 1.2$. It will be interesting to test the difference in approaches once we have a proper evaluation system.

After this, in order to finish the task, we only had to ask for some help on the necessary *GitHub Action* to finish the task. Thanks to the prompt support, that got finished smoothly.