Text Retrieval Using Natural Language Processing

Group 1C
Albert Anguera, Agnes Gubicza, Kanghwi Lee,
Modesto Orozco, Jakob Vokac

September 2, 2020

1. Part A

During the first part of the project, we experimented with BM25, sent2vec, elastic and hybrid search methods and M@L, MRR and ROUGE metrics.

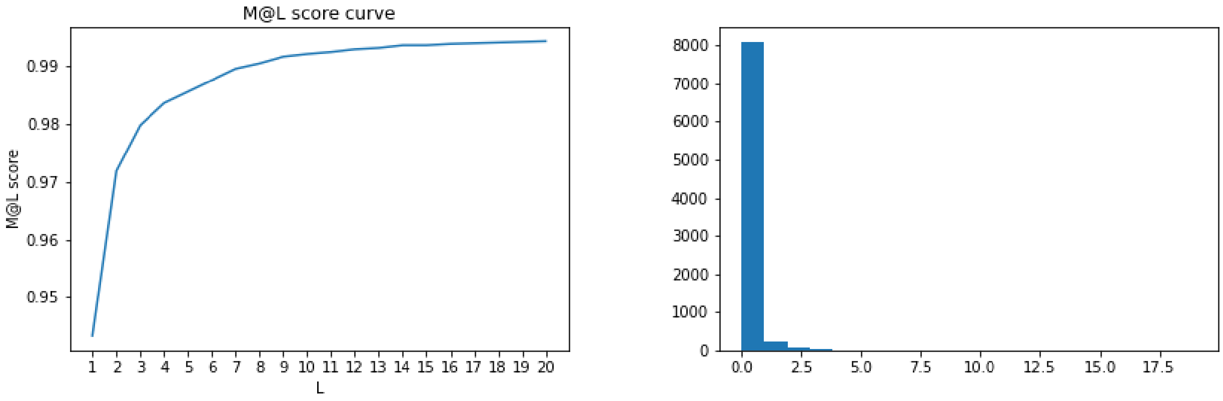
We worked with the reduced dataset of 10 000 papers, and reduced it even further to eliminate records with empty entries, such as papers with no abstract or title. We used the Natural Language Toolkit (NLTK), in particular the RegexpTokenizer.

Using title as the query and searching for the abstract, we summarized the results of sen2vec, hybrid and elastic search evaluated with M@L and MRR metrics in our mid-term report. Comparing the three search methods, the results can be interpreted as follows. Hybrid search with 'and' logic filtered out too many documents during boolean search, so it was the least accurate. Hybrid search with 'or' logic relaxed the boolean filter and gave a much more accurate result. Comparing hybrid-or search with sent2vec search, the latter gave a better M@20 score, but the former returned a higher M@1 and MRR score. We can interpret this as a result of boolean filtering. Boolean filtering reduced the number of documents to be searched per query, so if the target document was in the reduced set of 3 documents, it is more likely that the target document will be the closest, hence higher M@1 score. Elastic search has much more functions built into it, so it takes a bit more time than sent2vec or hybrid search, and returns the best result. However it is worth noting that elastic search is considered a very fast search engine when dealing with a huge amount of data.

Here we present results which were not included in the mid-term report.

BM25 ranking: Using titles as query items and matching abstracts, the predictions for the full filtered dataset took around 450 seconds. The M-score was performed over the first 20 results and was 0.9944. The M@L score curve follows a smooth increasing trend as we can see in Figure 1a and almost 8.000 (over 90%) of the abstracts are the most likely match for their real titles. Additionally, the MRR score for the BM25 searching engine was 0.9627.

ROUGE metric: Building on the BM25 search results, we can evaluate the distribution of the recall, precision and f1 metrics for the different ROUGE methods over each real pair title-abstract. As you can see in Figure 2, ROUGE-2 method is generally associated with lower performances in the 3 scorers, being more remarkable in the recall. It means that the overlap of bigrams (pairs of words) between the titles and abstracts is generally lower than the unigram (word by word). This is



(a) M@L score

(b) M distribution

Figure 1: Searching results for matching abstracts using the titles as query items.

something expectable, specially in long titles and abstracts because the probability of finding exact matching decreases with the size of the sequence (however, at the same time, the longer is the sequence, the less combinations are possible. For example, taking as reference ”the squared green table”, the four unigram options are: ”the”, ”squared”, ”green”, ”table” whereas the bigram options are just 3: ”the squared”, ”squared green”, ”green table”). Finally, ROUGE-L looks for the longest common subsequence which seems to be a unigram in the vast majority of the cases because the performances are very similar to the ones of ROUGE-1.

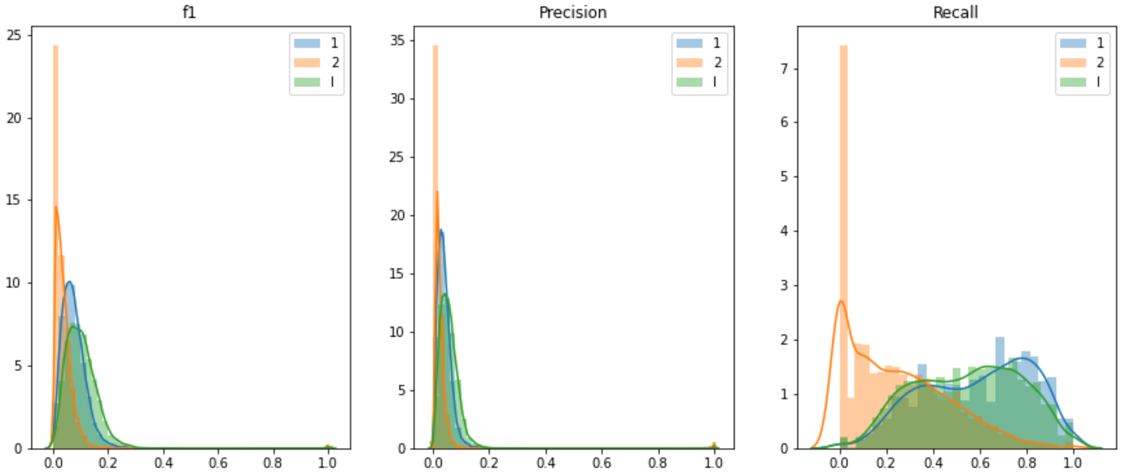


Figure 2: Rouge metrics evaluated on BM25 title to abstract queries

Figure 3 shows the mean values of the different metrics (using the different ROUGE methods) as a function of the position of the real abstract in the ranked relevance list. It means that, e.g. when taking the sixth title, we looked for the index of the sixth abstract in the ranked predicted list of abstracts. As we can see, the worse is the correspondence between title and abstract, the lower is the metric, meaning that f1, precision and recall in any of the ROUGE methods are sensitive to the BM25 predictions. Just by computing the f1, precision or recall of a certain pair title-abstract we could infer more or less accurately whether BM25 will work efficiently and and approximate position of the true abstract in the ranked predicted list.

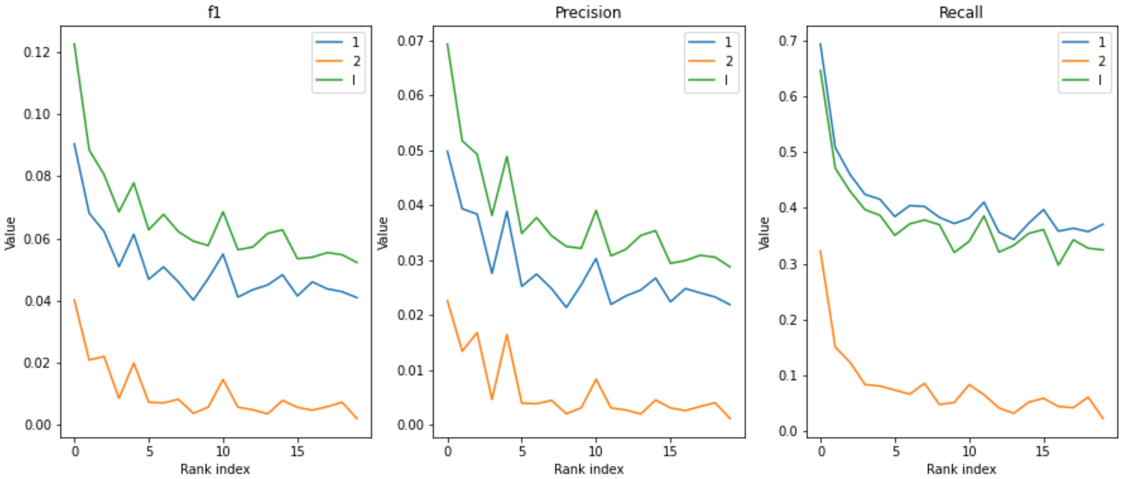


Figure 3: Mean values of the different ROUGE methods as a function of the position of the real abstract in the ranked relevance list

We also performed a BM25 ranking using the abstracts as the query term and looking for the corresponding fullbody, and evaluated the results with M@L, MRR and ROUGE. Due to space limitations, we do not include the figures here. The MRR score was 0.9961. Over 99% of the true full bodies were predicted as the first option and, if not, as the second. This makes unnecessary to compute the dependence of ROUGE metrics with the ranked positions. However, analysing their distributions we noticed that, for instance, precision takes generally high values (specially for ROUGE-1) meaning that the full body contains many of the abstract words.

BERT model: When researching BERT and transformer networks in general, we found that to code and train a fully functional BERT engine, we had two options. Either we take a pre-trained BERT model from Google or we train it ourselves on the PMC-OA dataset. The second option is expensive and would probably not yield good results with out resources. The first option is more manageable, however the problem lies in the limited vocabulary of the pre-trained BERT model. When BERT identifies a word not included in it's vocabulary it ignores it. Therefore, most of the technical and scientific terms, which are highly relevant for our search algorithms would get ignored. As such, we concluded that BERT was not a viable option for our search (embedding) methods.

2. Part B

2.1. Data preprocessing

2.1.1. Paper selection

We selected 10 papers from the Neural System course, randomly assigned 2 to each of the group members. All of us extracted the fullbody of the papers and picked 3 keywords to perform a hybrid search on the server and obtain 10 papers for each paper. We used the fullbody as the query. For keyword selection, we computed the TF-IDF score of the words in the document, but the first hits did not yield characteristic keywords. So we manually picked the keywords and used them with *and* or *or* logic. The 10 seed papers are listed in Appendix A. The corresponding code is in the notebook called Paper_query_and_paragraph_pairing.ipynb.

2.1.2. Paragraph pairing

Since the database had no information on the paragraph boundaries in the fulllbody of a paper, first we downloaded the pdf versions based on their DOI and split the fullbody to paragraphs accordingly. We identified the results and discussion paragraphs for all papers. In case of half of them we also paired the results and discussion paragraphs manually. We selected up to 3 discussion paragraphs per paper and identified the corresponding results paragraphs (max 3 result paragraphs per discussion paragraph).

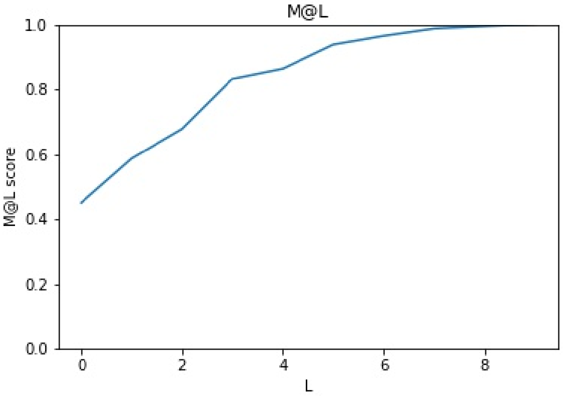
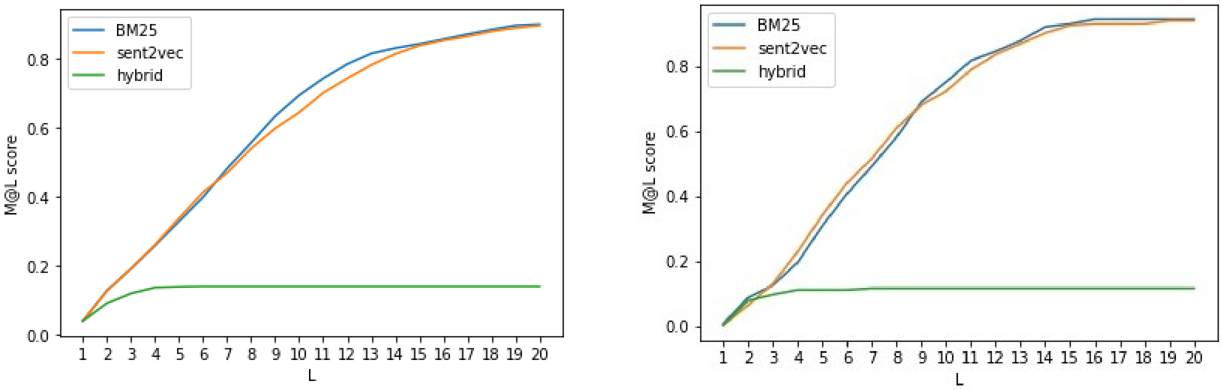


Figure 4: Comparison of manual and automatic BM25 paragraph pairing

Manual ranking is very cumbersome, we spent a significant amount of time on creating the paired dataset. Hence, we performed manual pairing on 50 papers and automatic pairing on the rest. Actually, we run the automatic pairing for all papers and compared the two methods. During autopairing, we paired all discussion paragraph with result paragraphs by ranking the paragraphs in the paper using BM25 as the metric (see auto_pairing.ipynb). Using the manual pairing as a ground truth, we calculated the M@L metric of the retrieved result paragraphs (Figure 4) in case of the 50 papers were both pairing methods were available. The BM25 pairing lists the correct paragraph with less than 50% chance, the corresponding result paragraph can be found among the first 8 results. The reason for this can be that the general keywords frequently appear throughout the whole paper, and BM25 is not able to capture the semantic content of each paragraph. The quality of the automatic pairing can be improved if we only rank result paragraphs.

2.2. Finding the corresponding discussion paragraph given results paragraph within the target paper

After creating our data set of paragraphs, we tried 3 types of text retrieval methods to find the corresponding discussion paragraph given the result paragraph, namely BM25, sent2vec embedding and hybrid search. We evaluated the performance of each with M@L, MRR (mean reciprocal rank), MAP, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and NDCG (normalized discounted cummulative gain) metrics. In this specific case, MRR and MAP gives the same scores, because we have one relevant document per query, so the average precision equals thee reciprocal rank. We calculated the average of ROUGE-1, 2 and l metrics for each query and used it as the relevance score for NDCG.



(a) M@L score of search methods using the full corpus **(b)** M@L score of search methods on the manually paired article.

Figure 5: M@L score of different discussion paragraph retrieval methods within the target paper given result paragraph

Our results are summarized in Figure 5. BM25 and sent2vec embedding performed very similar, while hybrid search lacked behind. We think the reason for this is that the keyword selection based on the TF-IDF is not optimal in a small data set and works much better in a larger corpus. BM25 and sent2vec were able to find reasonably well the corresponding discussion paragraph. We exploit the keyword selection in Section 2.4 in more detail.

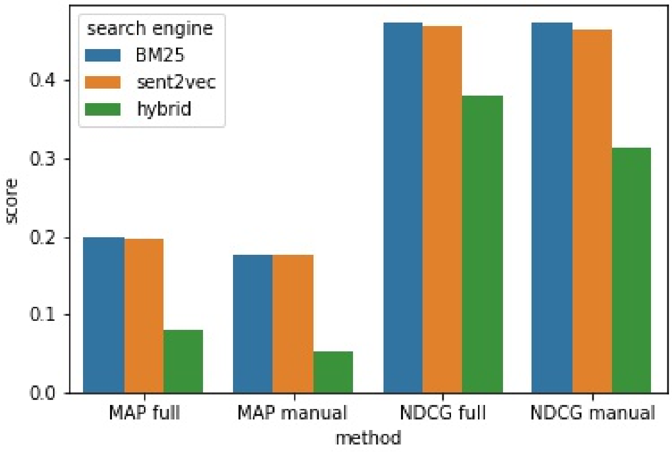
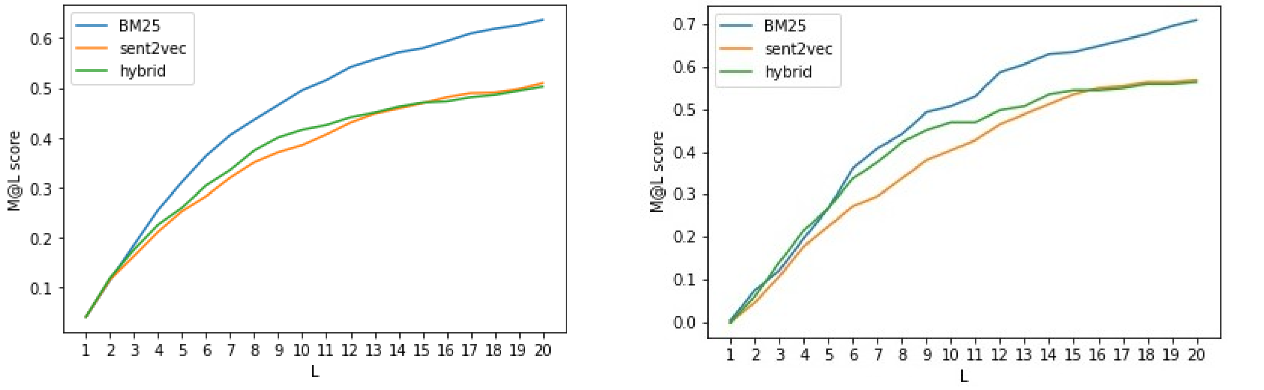


Figure 6: Metrics of retrieval methods querying within the paragraphs of the corresponding paper

BM25 and sent2vec shows very similar performance, not only in M@L score, but in other metrics as well (Figure 6). BM25 has a slightly increased score, but it is not statistically relevant. Hybrid search is very poor for the above described reasons.

2.3. Finding the corresponding discussion paragraph given results paragraph within all paragraphs in the data set

We performed the same search routines to find the corresponding discussion paragraph within all paragraphs of the data set. The metrics are also the same as described above. The results are plotted in Figure 7. The M@L score of hybrid search and sent2vec are very similar, while BM25 stands out. This can be for multiple reasons. On one hand, we paired half of the paragraphs automatically using BM25, which creates a bias in our dataset. On the other hand, we used sent2vec embedding out of the box and since it is a general purpose text-embedding method, it is not optimized to catch the semantic meaning of scientific publications. One improvement could be to train sent2vec on neurological papers, where we would need at least an order of magnitude larger dataset. The performance of the hybrid search is very sensitive to keyword selection, as it is investigated in Section 2.4.



(a) M@L score of different retrieval methods targeting the full data set.

(b) M@L score of search methods on the manually paired data set

Figure 7: Text retrieval performance over the full data set

The MAP and the NDCG scores also show the dominance of the BM25, and rate the performance of the hybrid search above the sent2vec embedding (Figure 8). This is an important factor, since after keyword selection, we use sent2vec embedding in the hybrid search as well. It shows that combining embedding based ranking with keyword selection can actually over-perform the embedding based ranking. The queries on the manually paired data set have slightly decreased scores compared to the full corpus. This can be because the BM25 pairing is closer to the applied search methods than manual pairing. It is an indication that the sent2vec embedding is not fully optimized on this subset of scientific papers. We have to note that our corpus is very small to draw strong conclusions from these scores.

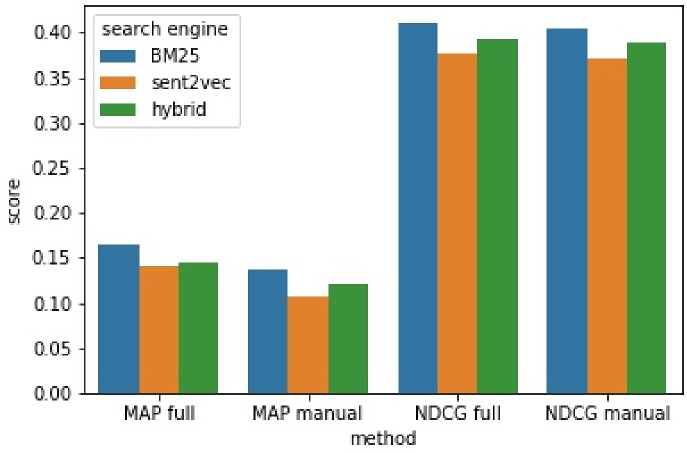
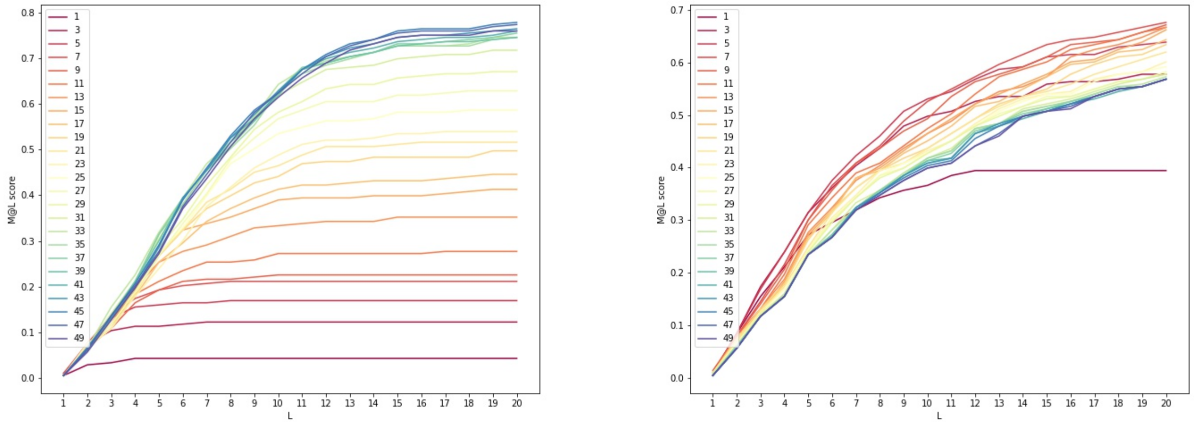


Figure 8: Metrics of retrieval methods querying within all paragraphs of the corpus

2.4. Understanding hybrid search engine performance

After evaluating the performance of three different searching methods (BM25, sent2vec, hybrid) in the last sections, we wanted to find out the origin of such a difference in the performance of our hybrid engine in the last 2 retrieval tasks. We explored, in the first part of the project, different combinations of the hybrid engine's hyper-parameters, concluding that ”or” logic and m=3 keywords was the optimum election. That combination was used in the second part of the project.
In the exercise of finding the corresponding discussion paragraph within all paragraph of the dataset, the hybrid engine worked reasonably well (slighly better than sent2vec), while when the dataset was reduced to the target article, the performance was not only worse, but dramatically poor. That was clearly indicative that the hybrid engine was not working correctly. Since the ”or” logic is less restrictive than ”and”, the only explanation for a bad functioning was in the selection of the keywords. We hypothesize that, in the second exercise (search restricted to the same article), since the corpus used to extract the relevant keywords was notably smaller, the election was not suitable and, therefore, performed bad for the chosen articles. If that was the case, increasing the number of keywords would improve the performance, up to a certain point, because the representative words would be more probably selected. In order to test it, we varied from m=1 to 50 the number of keywords chosen by the hybrid engine.



(a) M@L score of the hybrid search engine varying the number of keywords in the search of a discussion paragraph within the same paper.

(b) M@L score of the hybrid search engine varying the number of keywords in the search of a discussion paragraph in the full dataset.

Figure 9: Text retrieval performance using hybrid engine and different number of keywords

As we can see in Figure 9a, the election of m=3 was clearly inadequate for this exercise, where up to 45 keywords would be needed to perform an accurate search. Under that election, the performance would have been not only much better, but higher than in the full dataset search (bluish lines rise up to 0.8 in fig. 9a). This fits perfectly with our hypothesis. Additionally, we run the retrieval exercise over the full dataset of papers (fig. 9b) and observed that m=3 was clearly much appropriate. It is also interesting to notice that m=5 would have exploit all the capacity of the engine in that case.
Moreover, we can see that, for the large dataset, choosing from 3 to 30 keywords gives similar performances. However, the retrieval ability decreases when we keep increasing that number because of the introduction of noisy words. In turn, in the reduced dataset, the more words are used, the better the

performance (even though from 45 words, it seems to become steady).

To sum up, the hybrid engine performance is highly dependent on the number and quality of the chosen keywords. The smaller the dataset, the higher number of representative words need to be used because TF-IDF loses credibility. That is why in the retrieval exercise within the same paper, the election of 3 keywords is not the best option, whereas it is reasonable when using the full dataset.

Author contributions

We had regular online discussion where all team members took part. All of us equally contributed to the cumbersome paragraph pairing and fullbody splitting in Part B. Furthermore, we helped and supplemented each others codes.

Albert Anguera

Part A: Researched BERT
Part B: Worked on BM25 and sent2vec ranking to find discussion paragraph given results paragraph.

Agnes Gubicza

Part A: Implemented and performed elastic search (Elasticsearch_comparison_paragraphs.ipynb)
Part B: Worked on the details of paragraph pairing and dataset structuring and wrote the final report. Paper_query_and_paragraph_pairing.ipynb

Kanghwi Lee

Part A: Mid-term report. Implemented hybrid search, MAP and MRR metrics, which were also used in Part B.
Part B: Evaluation of the search results. Main coder for PartB_Withall_and_Within.ipynb (But everybody contributed to it)

Modesto Orozco

Part A: Implemented ROUGE metric and explored searching tasks.
Part B: Comparison of manual and automatic pairing. Worked on Bm25, sent2vec, elastic and hybrid search to find discussion par. given res. par. Study of hybrid search keyword dependence. Author of the auto_pairing.ipynb notebook.

Jakob Vokac

Part A: Researched BERT
Part B: Helped with coding in general, made and gave presentation.

A. Seed papers selected for querying the database

Albert:

Selective representation of relevant information by neurons in the primate prefrontal cortex.pdf

The Parahippocampal Place Area: Recognition, Navigation, or Encoding.pdf

Jakob:

Imaging systems level consolidation of novel associate memories.pdf

5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

Kanghwi:

Learning auditory discriminations from observation is efficient but less robust than learning from experience.pdf

A Specialized Forebrain Circuit for Vocal Babbling in the juvenile songbird.pdf

Modesto:

Convergent transcriptional specializations in the brains of humans and song-learning birds.pdf

Semantic memory and the brain: structure and processes.pdf

Agnes:

A model for memory systems based on processing modes rather than consciousness .pdf

Dynamic reorganization of striatal circuits during the acquisition and consolidation of a skill.pdf