Team Bruce Banner at Touché 2022: Argument Retrieval for Controversial Questions

Bernardo C. Moreira[1], Henrique Lopes Cardoso[1,2], Bruno Martins[3,4] and Fábio Goularte[3]

[1]*Faculdade de Engenharia da Universidade do Porto, Porto, Portugal*
[2]*Laboratório de Inteligência Artificial e Ciência de Computadores (LIACC), Porto, Portugal*
[3]*INESC-ID, Lisboa, Portugal*
[4]*Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal*

Abstract

Argument retrieval is a prominent topic in the context of current natural language processing applications. The task focuses on creating models that can retrieve coherent and strong arguments from textual sources. This technology can help individuals build an informed opinion about a controversial topic or support a particular stance on a debate. In this context, the Touché Task 1 was proposed within the scope of Conference and Labs of the Evaluation Forum 2022 (CLEF 2022), based on argument retrieval for controversial questions. We chose to compete with a sparse search. Despite ranking 3rd on relevance, 5th on quality and 4th on coherence, we concluded that our results are limited by our data arrangement process. The purpose of the task was to retrieve the most argumentative and relevant pairs of sentences, which could be formed with sentences from the same argument or not. Our approach focused on forming sentence pairs from the same argument, and achieved scores of 0.772 for quality, 0.651 for relevance, and 0.378 for coherence.

Keywords

Argument retrieval, Controversial questions, Sparse representation

1. Introduction

Currently, Natural Language Processing (NLP) is prevelant in almost everything done on the web. There are many research fields within the scope of NLP, such as Information Retrieval (IR), a process of accessing and retrieving the most pertinent information given a user query. More specifically, the focus of this work is on retrieving arguments for controversial topics.

Applications dealing with natural language use machine learning models for different purposes, such as speech analysis and understanding, generation of human-readable text, and information retrieval. The Touché lab[1], proposed in the scope of the Conference and Labs of the Evaluation Forum (CLEF), focuses on the task of retrieving arguments. The goal of the

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy
up201604014@fe.up.pt (B. C. Moreira); hlc@fe.up.pt (H. Lopes Cardoso); bruno.g.martins@tecnico.ulisboa.pt (B. Martins); fabio.goularte@inesc-id.pt (F. Goularte)
0000-0002-3307-0078 (B. C. Moreira); 0000-0003-1252-7515 (H. Lopes Cardoso); 0000-0002-3856-2936 (B. Martins); 0000-0003-4378-8734 (F. Goularte)

[1]https://webis.de/events/touche-22/

Touché lab is to motivate the NLP community to develop or improve upon existing technologies for argument mining and argument analysis. This year, this lab organized three shared tasks: Task 1 focused on argument retrieval for controversial questions; Task 2 on argument retrieval for comparative questions; and Task 3 focused on image retrieval for arguments.

Our team focused on the first task, where the goal is to retrieve and rank relevant sentence pairs from a collection of arguments, given a query about a controversial topic. Touché Task 1 at CLEF 2022 [1] is similar to the Touché Task 1 at CLEF 2021 [2], where given a query systems should retrieve relevant arguments. In contrast, in the 2022 edition, the participating teams have to retrieve a pair of sentences gathered from the arguments collection. This significant alteration increases the complexity of the task since the pair of sentences must be coherent, each sentence in the pair must be argumentative, and together the sentences should provide a summary of the argument from which they are retrieved. For example, if a user searches for "Libertarianism", an example of an output would be the pair of sentences "For example, Jim Babka, from Libertarian organization Downsize DC said they have had some very successful alliances with groups who would never describe themselves as conservative" and "Libertarians cooperate with many non right wing organizations".

The results obtained on preliminary experiments with data from the 2021 edition were encouraging; therefore, we decided to follow the same approach to the 2022 edition, which relies on Pyserini sparse search with BM25, further detailed in Section 3.

We have submitted four different runs for this shared task and leveraged the available evaluation files for the 2022 Task 1 to evaluate our approach. We achieved in our best run 0.772, 0.651, and 0.378 for Quality, Relevance, and Coherence scores, respectively. Human assessors manually define these metrics and evaluate the output's quality, relevance, and coherence, given a specific query. With these results we achieved third place in relevance, fourth in coherence and fifth in quality.

2. Background

Argument retrieval is a topic of growing interest. There have been some tested approaches in previous editions of this Touché task 1. As described in Bondarenko et al. [2], the usual approaches are based on DirichletLM [3] [4] and BM25 [5] models combined with WordNet [6] for query expansion.

BM25 is possibly one of the most used and essential functions in information retrieval and is used to estimate the relevance of documents to an input query. It is a non-linear combination of three document attributes: document frequency, document length, and term frequency.

Elrond, the team with the highest relevance score, utilized a DirichletLM-based retrieval model, using also the Krovetz stemming method [7], and excluded stop words. In the case of *Heimdall* [8], which was the team with the highest quality score, they utilized a DirichletLM but integrated with a topical relevance analysis using Universal Sentence Encoder and k-means clustering, and finally, a support vector regression model for argument quality trained on the Webis-ArgQuality-20 corpus[2].

[2]https://zenodo.org/record/3780049

Touché Task 1 2022 submissions were evaluated manually, by humam assessors, based on each submitted run's sentence pairs relevance and quality scores. This entails analyzing if each sentence is argumentative, if both sentences in the pair are coherent and whether this pair of sentences form a summary of the respective arguments.

3. Approach

We make use of Pyserini [9], a python toolkit for information retrieval search. Our approach focuses on Pyserini's sparse retrieval – via integration with Anserini IR toolkit built on the Lucene search library [10], both for indexing the data and performing the search, since it performs retrieval with BM25 ranking using bag-of-words representations.

We used last year's evaluation files since we didn't have this year's data to calculate our approach's quality and relevance scores. To assess our runs on these evaluation files, we used the trec_eval[3] tool which is the standard tool used by the TREC[4] community for evaluating a run, given the topics and the evaluation files. Figure 1 shows our approach composed by five steps: (1) Input, (2) Pre-processing, (3) Data, (4) Data processing, and (5) Output.

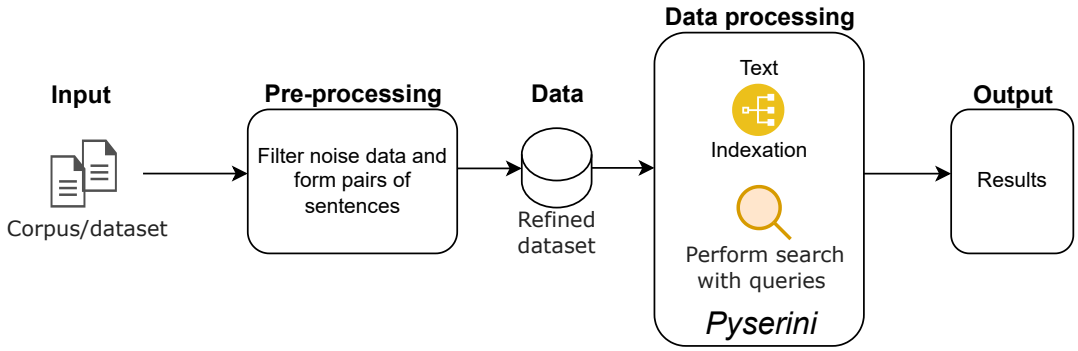

Figure 1: Our approach to the Touché Task 1 problem.

Our team submitted four different runs. All these runs were generated using Pyserini sparse representations, given the good results in the experiments performed on last year's data. All four runs submitted were generated using two variations of data and two variations of queries, as described in Section 3.3.

Since there was no available method to test the approaches before submitting, we needed to test the planned strategies with the available data from last year's task. With this in mind, we performed a series of experiments using the quality and relevance data available which provided an overview of our approaches.

The data was organized in two forms to evaluate the best arrangement of data for this task. The argument layouts tested were: (PC) Conclusions and Premises concatenated and (PCT)

[3]https://github.com/usnistgov/trec_eval
[4]https://trec.nist.gov/

Conclusions, Premises, and Topic concatenated.

In Table 1, our team results for each argument layout are compared with the top 3 results from last year's edition, both for quality and relevance. We noticed that using premise and conclusion concatenated provided a better quality score than premise conclusion and topic, but using the latter results in a slightly better relevance score. This occurs because introducing the topic sentence to the premise will negatively affect the semantic quality of the sentence since these sentences might not be perfectly aligned. On the other hand, concatenating this topic adds information to the premise; this means that it provides more details on the discussion topic the sentence is inserted in, resulting in a better relevance score.

Table 1
Quality and relevance NDCG@5 scores for our approaches, and to the top 2 teams from last year's competition.

Approach	Quality	Relevance
PC	**0.8116**	0.6742
PCT	0.7881	**0.6838**

Team Name	Quality
Heimdall	**0.841**
Skeletor	0.827

Team Name	Relevance
Elrond	**0.720**
Pippin Took	0.705

3.1. Input

The data provided for this year's task is separated into two different files. The first one is the data to be used for retrieval, and the other one contains the queries.

Each element in the data file is an *argument* containing five fields: (1) *id*, (2) *conclusion*, (3) *premises*, (4) *context*, and (5) *sentences*.

The sentences field is used for this task – the aim is to retrieve a pair of sentences formed with sentences from the same or different argument. The *sentences* field aggregates the *conclusion* of the argument and each *sentence* in its premises, each identified by a unique *id*.

The queries file contains the *topic id*, the *title* which is the question itself, the *description*, and the *narrative*. The description provides an explanation and a context for the given question, while the narrative describes the desired outcome.

3.2. Pre-Processing

Our team chose to follow the above mentioned and tested approach to the 2021 Task 1, which relies on pyserini sparse to index and perform the search of relevant documents for a given query. As it was already mentioned, this approach consists of five steps: (1) Input, (2) Pre-processing, (3) Data, (4) Data processing, and (5) Output, as show in Figure 1. The data processing phase consists in generating the data in a format amenable for an information retrieval setup. This task requires that a query's output be formed as a pair of sentences. The "sentences" field in the argument contains all the sentences and respective id; therefore, we processed this field to develop all possible pairs of sentences from the same argument.

For example, the argument with id **Sf9294c83-A9a4e056e** contains three sentences: two premises and one conclusion. After being processed, these sentences generated three different pairs of sentences (see Figure 2).

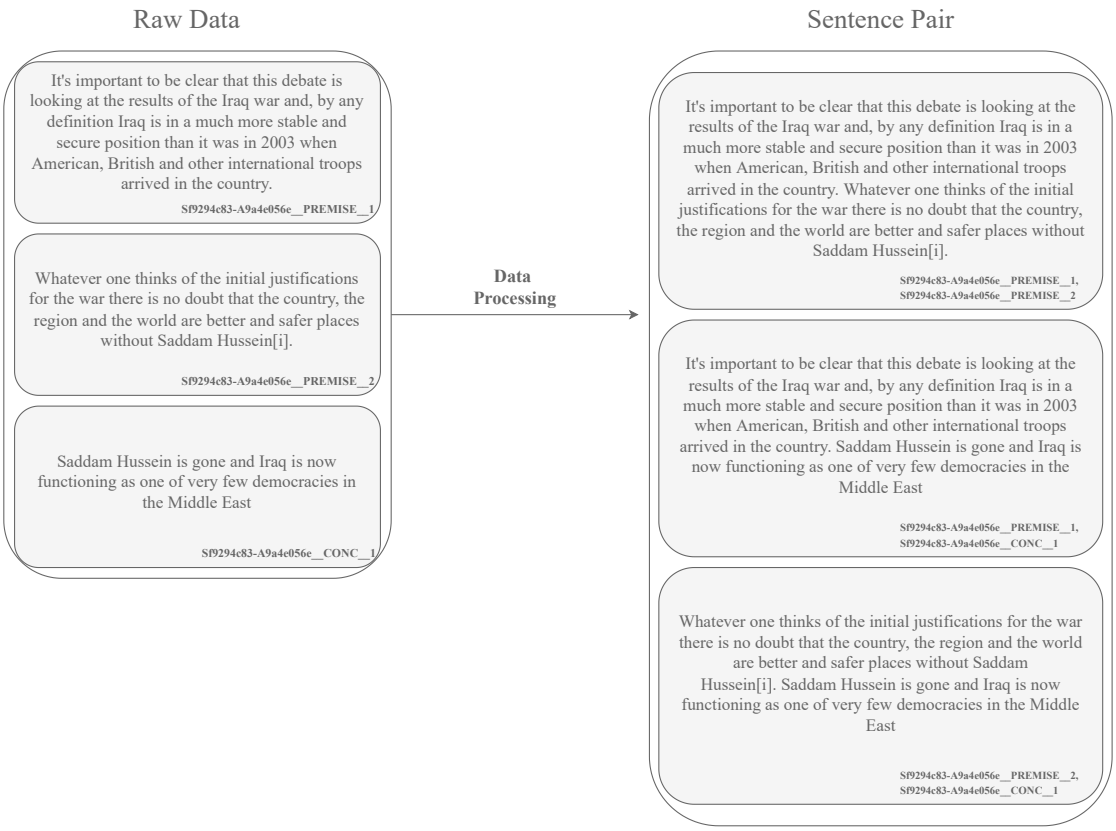

Figure 2: Processing raw data to form pairs of sentences. Each sentence contains an id and content; both the sentence id and the sentence content are concatenated to form the sentence pair. The number of pairs formed depends on the number of sentences in the array.

3.3. Data

The query data (see Listing 1) was used in two ways. The first and more straightforward consists of using only the query. The second, an expanded query version, includes the query concatenated with the narrative and the description.

```
<topic>
  <number>1</number>
  <title>Should teachers get tenure?</title>
  <description> A user has heard that some countries do give teachers tenure and others don't. Interested in the
      reasoning for or against tenure, the user searches for positive and negative arguments. The situation of
      school teachers vs. university professors is of interest. </description>
  <narrative> Highly relevant arguments make a clear statement about tenure for teachers in schools or
      universities. Relevant arguments consider tenure more generally, not specifically for teachers, or, instead
      of talking about tenure, consider the situation of teachers' financial independence. </narrative>
</topic>
```

Listing 1: Example of a topic in the query data.

The new generated data, which will be indexed by Pyserini, is composed of documents containing each an id and a sentence pair.

Argument sentences were paired following two different arrangements, as shown in Figure 3.

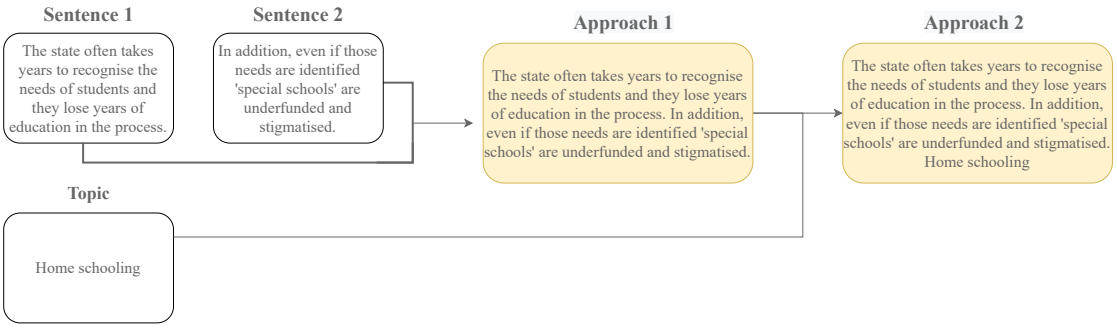

Figure 3: Representation of the two different arrangements of sentences used.

The first approach consists in using both sentences concatenated, while the second approach concatenates both the sentences and the topic.

As shown in Figure 4, with these different approaches for both queries and sentence pairs, we submitted four different runs:

- pyserini_sparse_v1: pairs of sentences, and query with title, description and narrative
- pyserini_sparse_v2: pairs of sentences, and query with title only
- pyserini_sparse_v3: pairs of sentences with topic concatenated, and query with title, description and narrative
- pyserini_sparse_v4: pairs of sentences with topic concatenated, and query with title only

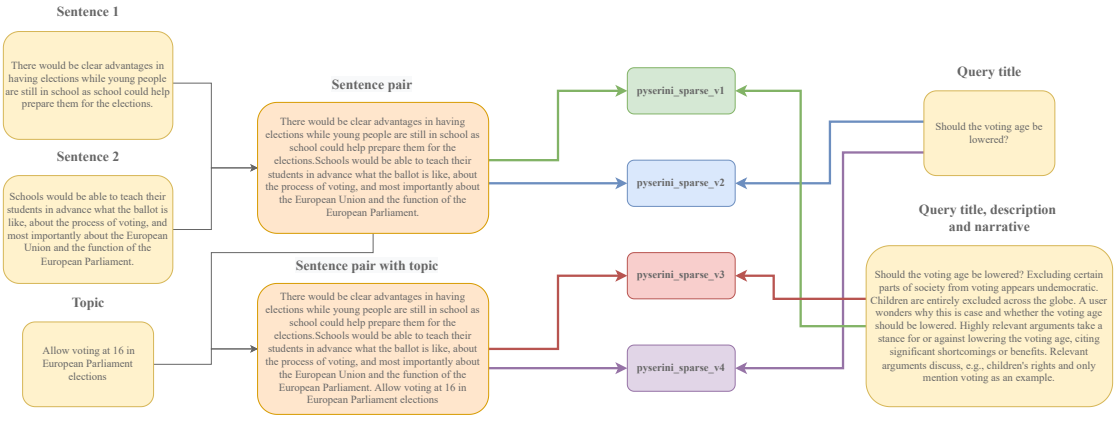

Figure 4: Representation of the four combinations of data used with an example.

3.4. Data Processing

The Pyserini toolkit, which is responsible for indexing the new data and performing the search with the given queries, allows a variety of approaches. Given our time and resource limitations, the best approach to this task is to use sparse representations. Pyserini sparse representations produces smaller indexes than dense search. Pyserini also includes a hybrid approach that makes use of both sparse and dense indexes; however, given our approach that generates a high number of sentence pairs, we did not generate dense representations for our sentence pairs, and thus do not make use of the hybrid approach.

The Pyserini toolkit receives the data as an input, which it indexes and uses to search for documents given a query. Both the index and search processes use the Lucene library; for the search process, we specified how many hits for each question we need (1000) and which scoring function to use, which in our case is BM25 with default values from Pyserini.

3.5. Output

The post-processing step consists of cleaning Pyserini's output to match the structure requested for Touché Task 1, which is the standard TREC format:

qid stance pair rank score tag

Even though the format of the document, created in the previous step, is the correct one, our data is not ready to submit. In this step, to provide all the required data for the submission, we iterate through the results file to replace the default values for stance (*Q0*) and tag (*Anserini*) produced by Pyserini. These values are replaced with the correct stance of the pair of sentences and our team tag *Bruce-Banner*, respectively, resulting in a new and valid results file.

4. Experimental Evaluation

After the release of Touché Task 1 2022 evaluation files we performed experiments to evaluate the performance of our runs (see Table 2). The evaluation files contained the top pairs of sentences, for each query, ranked according to their general topical relevance, their argument quality and their coherence. Eleven teams participated in this Task, and according to these metrics, our team **Bruce-Banner** was placed in third in relevance, fourth in coherence and fifth in quality, as shown in Table 3.

Table 2
Quality, relevance and coherence results for our four runs submitted in Touché Task1 2022.

Run	Quality NDCG@5	Relevance NDCG@5	Coherence NDCG@5
pyserini_sparse_v1	**0.772**	0.641	**0.378**
pyserini_sparse_v2	0.701	0.580	0.353
pyserini_sparse_v3	0.760	**0.651**	0.354
pyserini_sparse_v4	0.709	0.586	0.357

Table 3
Touché Task 1 2022 Quality, Relevance and Coherence teams scores.

Team	Relevance NDCG@5
Porthos	0.742
Daario Naharis	0.683
Bruce Banner	**0.651**
D Artagnan	0.642
Gamora	0.616
Hit Girl	0.588
Pearl	0.481
Gorgon	0.408
General Grievous	0.403
Swordsman	0.356
Korg	0.252

Team	Quality NDCG@5
Daario Naharis	0.913
Porthos	0.873
Gamora	0.785
Hit Girl	0.776
Bruce Banner	**0.772**
Gorgon	0.742
D Artagnan	0.733
Pearl	0.678
Swordsman	0.608
General Grievous	0.517
Korg	0.453

Team	Coherence NDCG@5
Daario Naharis	0.458
Porthos	0.429
Pearl	0.398
Bruce Banner	**0.378**
D Artagnan	0.378
Hit Girl	0.377
Gamora	0.285
Gorgon	0.282
Swordsman	0.248
General Grievous	0.231
Korg	0.168

Even though our team placed third in relevance, fourth in coherence, and fifth in quality, we believe that our approach would benefit from a different data arrangement process. In our approach, we chose to form all possible pairs of sentences within the same argument, but sentences from different arguments were also accepted for the task. However, in the quality evaluation file provided by the task, sentence pairs formed with sentences from different arguments vary between 35.6% and 46.9% across topics. Still, while our approach does provide sentence pairs that have been well-ranked, there is a limitation on the provenance of pairs we were able to form, which directly determines the retrieved entries.

5. Conclusions

Our team decided to participate in this Task with a sparse search approach. Our data processing stage influenced our approach. For this Task, it was allowed and even desired to form pairs with sentences from different arguments, but we decided to form all the possible pairs of sentences from the same arguments. Thus, various pairs of sentences were not indexed, resulting in a poorer collection of documents to retrieve.

As future work it would be interesting to:

- Form data with pairs of sentences, including pairs from different arguments. Sentence selection needs to be done carefully; otherwise, if all the possible pairs of sentences are formed, the resulting collection will be enormous, which may lead to the impossibility of using pyserini search or bring enormous computational costs;
- Follow a different approach resulting in lesser data so that pyserini hybrid retrieval can be used, i.e., using pyserini sparse to retrieve the top 2000 pairs of sentences and then using pyserini hybrid to re-rank those retrieved pairs;
- Re-rank the arguments returned with a similar score using a fine-tuned BERT model to assess argument quality [11].

It would also be interesting to analyze the other teams' approaches, such as *Porthos* and *Daario Naharis*, which achieved scores better than ours.

References

[1] A. Bondarenko, M. Fröbe, J. Kiesel, S. Syed, T. Gurcke, M. Beloucif, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, M. Potthast, M. Hagen, Overview of Touché 2022: Argument Retrieval, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. 13th International Conference of the CLEF Association (CLEF 2022), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2022, p. to appear.

[2] A. Bondarenko, L. Gienapp, M. Fröbe, M. Beloucif, Y. Ajjour, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, M. Potthast, M. Hagen, Overview of Touché 2021: Argument Retrieval, in: K. Candan, B. Ionescu, L. Goeuriot, H. Müller, A. Joly, M. Maistro, F. Piroi, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. 12th International Conference of the CLEF Association (CLEF 2021), volume 12880 of *Lecture Notes in Computer Science*, Springer, Berlin Heidelberg New York, 2021, pp. 450–467. URL: https://link.springer.com/chapter/10.1007/978-3-030-85251-1_28. doi:10.1007/978-3-030-85251-1_28.

[3] D. J. MacKay, L. C. B. Peto, A hierarchical dirichlet language model, Natural language engineering 1 (1995) 289–308.

[4] J. M. Ponte, W. B. Croft, A language modeling approach to information retrieval, in: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '98, Association for Computing Machinery, New York, NY, USA, 1998, p. 275–281. URL: https://doi.org/10.1145/290941.291008. doi:10.1145/290941.291008.

[5] S. Robertson, H. Zaragoza, The probabilistic relevance framework: Bm25 and beyond, Foundations and Trends® in Information Retrieval 3 (2009) 333–389. URL: http://dx.doi.org/10.1561/1500000019. doi:10.1561/1500000019.

[6] C. D. Fellbaum, Wordnet : an electronic lexical database, Language 76 (2000) 706.

[7] R. Krovetz, Viewing morphology as an inference process, Artificial intelligence 118 (2000) 277–294.

[8] L. Gienapp, Quality-aware argument retrieval with topical clustering, Working Notes of CLEF (2021).

[9] J. Lin, X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep, R. Nogueira, Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations, in: Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021), 2021, pp. 2356–2362.

[10] E. Hatcher, O. Gospodnetic, Lucene in action (in action series), Manning Publications Co., 2004.

[11] S. Gretz, R. Friedman, E. Cohen-Karlik, A. Toledo, D. Lahav, R. Aharonov, N. Slonim, A large-scale dataset for argument quality ranking: Construction and analysis, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 7805–7813.