Thor at Touché 2021: Argument Retrieval for Comparative Questions

Notebook for the Touché Lab on Argument Retrieval at CLEF 2021

Ekaterina Shirshakova[1], Ahmad Wattar[2]

[1]Martin Luther University Halle-Wittenberg, Germany
ekaterina.schirschakova@student.uni-halle.de
[2]Martin Luther University Halle-Wittenberg, Germany
ahmad.wattar@student.uni-halle.de

Abstract

We report on our approach for the Shared Task 2 from the second Touché lab [1] on argument retrieval at CLEF 2021. We retrieved and ranked the documents from ClueWeb12 to answer comparative questions using Okapi-BM25. We supplied the ranking model with 2 different kinds of expansion—query expansion with synonyms from WordNet and index expansion with arguments from Targer. Examination of different combination of expansions and search fields showed, that although the expansions benefit only one field while cause worse results for another, in case of synonyms it is still more beneficial to use both affected fields together with the expansion. We further re-rank the initial set of document candidates using elasticsearch and it's built-in BM-25 algorithm.

1. Introduction

People always feel the need to compare things with each other, when they have to make a decision about which one to choose. This comparison can be used to find out the advantages and disadvantages of certain objects, or simply to highlight the differences between them. In everyday life this need can be fulfilled by search engines such as Google, Yahoo, DuckDuckGo, Yandex and many others, as one can just type in their questions, i.e. "What is better Linux or Windows?", in a search field. It became even more convenient with the use of various voice assistants, as one can simply ask such a question.

However comparative queries are a very specific type of queries, that demand specific solutions. Such queries can contain an explicit comparison between two or more objects (like in an example above) or even implicitly compare all entities within a more general group of entities (i.e. a question "Who is the best singer in the USA?" implies comparison between all singers in the USA). Returning documents that simply contain information about just one of the compared objects would not be enough to meet the need of comparison between them. As this problem drives attention of many researchers, such events as the Touché Lab on Argument Retrieval

CLEF 2021, 21-24 September 2021, Bucharest, Romania
ekaterina.schirschakova@student.uni-halle.de (E. Shirshakova); ahmad.wattar@student.uni-halle.de (A. Wattar)
https://github.com/eshirshakova (E. Shirshakova); https://github.com/ahmad-Wattar (A. Wattar)
0000-0002-9026-8806 (E. Shirshakova); 0000-0002-3434-566X (A. Wattar)

[1, 2] at CLEF 2021 are a good starting point for those who want to find a practical solution for it.
During the winter semester 2020/2021, we dealt with the Shared Task 2 of the Touché Lab on Argument Retrieval at CLEF 2021. The task consisted of retrieving documents from ClueWeb12 for 50 different topics and then indexing these documents to develop a search engine that returns the most relevant results. The topics were provided by the organisators and formulated as comparative queries, for example "Which is better, a Mac or a PC?".
For retrieving the documents we used the API of *ChatNoir* [3] along with the python library *requests*. For each query we have saved the first 10,000 documents that were returned by the *ChatNoir*. However, after various experiments we decided to build an index for only the first 110 documents retrieved from *ChatNoir* for each query, because *ChatNoir* already delivers mainly relevant documents within the first 110 results. For indexing we decided to use *elasticsearch* – the same open-source instrument that was used to improve *ChatNoir* [4]. Before building the index, we removed punctuation marks and lemmatized verbs and nouns with *nltk*. We also experimented with *TARGER* [5] and *WordNet* [6]. We found out, that saving arguments retrieved with *TARGER* along with the document or query expansion with synonyms from *WordNet* can provide a slight improvement of results for one search field, but at the same time less relevant documents will be returned when using another search field with the same extension.
At the end, we evaluated our experiments with ndcg10, recall10, precision and F1-score and saved our results in a table.

2. Related Work

The interest of natural language processing and especially comparative question answering is increasing because of evermore growing use of natural language technologies in an everyday life – many companies develop or use chatbots, interactive bots at call-centers, programs that can analyze documents and many other useful tools that demand computers to understand natural language. There is a number of researchers engaged in research of argumentation mining, for example, Wachsmuth et al. [7] developed an argument search framework using a composition of approaches for acquiring, mining, assessing, indexing, querying, retrieving, ranking, and presenting arguments. Stab et al. [8] also introduced an argument search system. This system allows argument searching in heterogeneous texts. Another approach for argument retrieval within massive unstructured corpora was proposed by Levy et al. [9]. Here increasing of potential coverage of documents retrieved for queries was achieved by query relaxation.

As a part of argument mining problem, claim and premisse detection also drives attention of various researchers: for example, Ajjour et al. [10] refers to unit segmentation in order to enhance relevance of documents containing arguments for search engines. Levy et al. [11] proposes automatic claim detection solution, which can be used on large coropora.

Eger et al. [12] see argumentational mining as a token-based dependency parsing and a token-based sequence tagging problem, hence use a multi-task learning setup to solve it. The neural argument mining framework TARGER, proposed by Chernodub et al. [5], was used in our approach, as it is an open-source solution which can be used in real-time. Huck [13] also

used TARGER in the elasticsearch-based search engine developed for the Shared Task 2 from Touché lab on argument retrieval at CLEF 2020. As his approach showed promising results, we decided to explore possibilities of further improvement of his approach in our work.

3. Approach

3.1. *Document Retrieval*

To build and test the search engine, we use the topics published for the Shared Task 2 from Touché lab on argument retrieval at CLEF 2020. We first convert input data (queries with description and narrative) from xml to json by using *xml.etree.ElementTree* python module.
To retrieve documents from *ChatNoir*, we use the python library *requests*, which allows us to work with *ChatNoir* API. We use the operator "AND" for each query, as after visual inspection of the first 10 results we found that it tends to deliver more relevant and useful results compared to an operator "OR". We also remove punctuation, as punctuation marks are considered as noise, and it is a common practice in information retrieval. We use the python library *boilerpy3* to extract the content of each found document.
At the end for each topic we saved a dictionary containing keys 'number', 'title', 'description', 'narrative', 'results' (number of documents found in *ChatNoir*) and 'documents' (list of documents retrieved from *ChatNoir*). We save each dictionary in the json-format to a separate file.

3.2. *Indexing*

First we rearrange the data to make it more convenient for indexing. We read the queries previously saved as dictionaries and iterate over the first 110 documents saved for each query. For each document we create a lemmatized version, using *nltk*. We only lemmatize nouns and verbs, as comparative adjectives are important for comparison. We use the same technique to lemmatize the title. Lemmatization helps to find words regardless of their form in the text. We wanted to test at least two different improvements (query expansion with synonyms and adding arguments to search fields), because we wanted to compare the performance of different approaches. In order to do so, at this step we also try to retrieve arguments: we send the unlemmatized document to *TARGER* by using the same library *requests* we used previously for retrieving documents from *ChatNoir*. We decided to use the *Combo*-model of *TARGER*, because as a combination of various other models it tends to deliver the best result. After visual inspection of retrieved arguments for several random topics we decided to save the tokens, that received a "P" or "C"-label (which means, were detected as premise or claim) with probability higher than 0.99. The most tokens with less probability for the same labels aren't parts of an argument. We lemmatize arguments in the same way we did it for the document and its title.
As we preprocessed all text fields needed to be saved to index, we construct bulk data which then will be used to create an *elasticsearch* index. For the body of each index element we create a dictionary containing query, title, lemmatized title, topic number, uuid of the document, its relevance score from *ChatNoir*, the document itself, lemmatized document and arguments retrieved from it. All those keys can be further used as search fields in the index, although for

our purposes we only used the lemmatized versions of the document, title and arguments. Other fields are either used to represent a document in a convenient for the user form (the original document and title) or to further evaluation (other fields). We create an *elasticsearch*-index from bulk data and adjust BM-25 parameters. Our final parameters are b=0.68 and k1=1.2, as they showed the best results for an ndcg@10-score for the topics and relevance judgements from Touché Lab 2020.

3.3. *Query Expansion*

For query expansion we decide to use synonyms retrieved from *WordNet* [6], because it can help to retrieve documents with relevant terms that were implied by the user, but weren't mentioned explicitly (as queries in a natural language usually don't contain synonyms for the words used in the query: it is considered as unnecessary duplication). We save lemmatized topics to a dataframe with 2 columns: 'query' and 'syn' for synonyms. We then try to find synonyms for words from the query in order to cover a higher variety of possible options. For this purpose we collect synonyms for separate words from *WordNet*. We first tried to manually detect comparison objects for each query and only find synonyms for them. However, as participating in the Shared Task demands submitting your approach to TIRA platform [14], where it will be tested on a virtual machine, the manual selection wasn't possible. Hence after experimenting with only synonyms for objects, we have selected the setup with the best results and for this setup we collect synonyms for all words in the query. We remove duplicates from the synonyms found and save them in the 'syn' column. Then the 'syn' column will be used to expand the original query.

3.4. *Ranking*

Okapi-BM25 is the default ranking function of *elasticsearch*. As we could get decent results already for default parameters with this function, we decided to keep it. For adjusting the ranking for the topics from 2020, we tried to change the parameter b with an 0.01 step in a range from 0.77 to 0.65. The best option is b=0.68. We link it with the fact that many articles are dedicated to a specific topic, thus the longer documents shouldn't be considered less relevant just because of their length (which would be the opposite, if one document covers many topics: then the longer one would probably contain more topics than needed and because of it be less relevant). We also tried different numbers for the k1 parameter, which stands for term frequency saturation. The default version of k1=1.2 showed the best result, probably because the most documents retrieved are articles from various web-sites (the corpus isn't dedicated to a specific range of topics, and the average length of the documents isn't very long, what would be the case for the books, or very short, what would be the case for a corpus consisting of Twitter posts).
In order to get the best results, we explored combinations of different searching fields along with adding and removing the query expansion with synonyms. As search fields, we decided to take lemmatized title, lemmatized document and arguments in various combinations: as single search fields, all possible pairs and finally all three fields. We also tried all above mentioned combinations with adding synonyms to each query.

We achieved the best results, when we took lemmatized document and lemmatized title as search fields along with the weighted query expansion with synonyms (we set the boost parameter to 5 for the main query; this is similar to weight=1 for the main query and weight=0.2 for the synonyms). As for the arguments search field, used alone it showed the worst results, because for many documents no arguments from TARGER [5] weren't delivered.

4. Evaluation

The task of Touché Lab on Argument Retrieval at CLEF 2021 also includes the evaluation of the developed search engine. To adjust parameters and compare different options, we used topics and relevance judgements from Touché 2020 [15]. We then have deployed our search engine with the parameters that showed the best results to the TIRA rating platform [14]. In TIRA, each participant in the task receives their own virtual machine and sends a retrieval model to the used task.

4.1. Experimental setup

We used 4 different evaluation metrics to evaluate our approach, namely ndcg@10, recall@10, precision@10 and F1-score. With the help of these 4 evaluation metrics, we analyzed the results of all combination mentioned above: topic, title and arguments search fields in all possible combinations with and without query expansion. We achieved the best results when using document and title as search fields with query expansion.
We have then adjusted the parameters (b and k1) of the BM-25 for search in documents and titles. As you can see in table 1, we achieved the best ndcg@10 value when setting b to 0.68 and k1 to 1.2.

Table 1
Adjustment of BM-25 parameters for documents and titles

b	k1	Ndcg@10
0.75	1.2	0.4404
0.74	1.2	0.4394
0.72	1.2	0.4405
0.71	1.2	0.4405
0.70	1.2	0.4411
0.69	1.2	0.4415
0.68	**1.2**	**0.4434**
0.68	1.21	0.4427
0.68	1.19	0.4433
0.8	1.3	0.4348
0.7	1.1	0.4396
0	5	0.3372

5. Results

Table 2 represents the evaluation values for each experiment with search fields and query expansion for the topics used for the Shared Task 2 in 2020. Our approach achieves the best ndcg@10 value when we expand the objects in the queries with synonyms and use document and title as search fields. Using this option for the Shared Task 2 in 2021, we achieved the following results: ndcg@5 for relevance 0.478, for quality 0.680.

Table 2
For each search field (doc, title, args) we evaluate all possible combinations (doc, title and args alone, then pairs: doc+title, args+title, doc+args) and all search fields together (doc+title+args). For each option we evaluate searching with no extension (basic) and adding synonyms to the query (syn).

Approach	Ndcg@10	Recall@10	Precision	F1-score
doc				
syn	0.3236	0.1990	0.300	0.2393
basic	0.3182	0.1962	0.292	0.2347
title				
syn	0.3482	0.2187	0.340	0.2662
basic	0.3287	0.2056	0.312	0.2479
args				
syn	0.2704	0.1810	0.260	0.2134
basic	0.2646	0.1801	0.262	0.2135
doc+title				
syn	**0.4450**	**0.2800**	**0.428**	**0.3385**
basic	0.4434	0.2716	0.422	0.3305
args+title				
basic	0.3863	0.2531	0.376	0.3025
syn	0.3792	0.2503	0.370	0.2986
doc+args				
basic	0.2852	0.1898	0.276	0.2249
syn	0.2813	0.1861	0.274	0.2216
doc+title+args				
basic	0.3768	0.2344	0.352	0.2814
syn	0.3669	0.2298	0.348	0.2768

6. Conclusion

We found out, that using arguments extracted from TARGER for the title search field can provide slight improvement, while using them with the document search field decreases ndcg@10 as well as the other evaluation metrics. Possible reason could be increased term frequency in the document for query terms (as arguments are some sentences from the document), which represents lesser significance of the query terms in BM-25. On the other hand, arguments increase relevance when using title alone as search field. This fact could be used for further improvement of the search engine. We assume, that including tokens with lesser probability

score for premise and clause from TARGER along with using title and arguments as search fields could lead to better results for this pair of search fields; however, we still think that this approach wouldn't be better than using document and title as search fields together with the query expansion. Although we think that it is a hypothesis worth testing.
As using both fields with adjusted BM-25 parameters show significant improvement compared to use of single search fields, query expansion with synonyms for this task was a preferable option over adding arguments to the index. In general, using synonyms to expand comparative queries seems to be a promising direction.

References

[1] 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: D. Hiemstra, M.-F. Moens, J. Mothe, R. Perego, M. Potthast, F. Sebastiani (Eds.), Advances in Information Retrieval. 43rd European Conference on IR Research (ECIR 2021), volume 12036 of *Lecture Notes in Computer Science*, Springer, Berlin Heidelberg New York, 2021, pp. 574–582. URL: https://link.springer.com/chapter/10.1007/978-3-030-72240-1_67. doi:10.1007/978-3-030-72240-1_67.

[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: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), Working Notes Papers of the CLEF 2021 Evaluation Labs, CEUR Workshop Proceedings, 2021.

[3] M. Potthast, M. Hagen, B. Stein, J. Graßegger, M. Michel, M. Tippmann, C. Welsch, ChatNoir: A Search Engine for the ClueWeb09 Corpus, in: B. Hersh, J. Callan, Y. Maarek, M. Sanderson (Eds.), 35th International ACM Conference on Research and Development in Information Retrieval (SIGIR 2012), ACM, 2012, p. 1004. doi:10.1145/2348283.2348429.

[4] J. Bevendorff, B. Stein, M. Hagen, M. Potthast, Elastic ChatNoir: Search Engine for the ClueWeb and the Common Crawl, in: L. Azzopardi, A. Hanbury, G. Pasi, B. Piwowarski (Eds.), Advances in Information Retrieval. 40th European Conference on IR Research (ECIR 2018), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2018.

[5] A. Chernodub, O. Oliynyk, P. Heidenreich, A. Bondarenko, M. Hagen, C. Biemann, A. Panchenko, TARGER: Neural argument mining at your fingertips, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Florence, Italy, 2019, pp. 195–200. URL: https://www.aclweb.org/anthology/P19-3031. doi:10.18653/v1/P19-3031.

[6] G. A. Miller, Wordnet: A lexical database for english, Communications of the ACM 38 (1995) 39–41. URL: http://doi.acm.org/10.1145/219717.219748. doi:10.1145/219717.219748.

[7] H. Wachsmuth, M. Potthast, K. Al-Khatib, Y. Ajjour, J. Puschmann, J. Qu, J. Dorsch, V. Morari, J. Bevendorff, B. Stein, Building an argument search engine for the web, in: Proceedings of the 4th Workshop on Argument Mining, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 49–59. URL: https://www.aclweb.org/anthology/W17-5106. doi:10.18653/v1/W17-5106.

[8] C. Stab, J. Daxenberger, C. Stahlhut, T. Miller, B. Schiller, C. Tauchmann, S. Eger, I. Gurevych, ArgumenText: Searching for arguments in heterogeneous sources, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 21–25. URL: https://www.aclweb.org/anthology/N18-5005. doi:10.18653/v1/N18-5005.

[9] R. Levy, B. Bogin, S. Gretz, R. Aharonov, N. Slonim, Towards an argumentative content search engine using weak supervision, in: COLING, 2018.

[10] Y. Ajjour, W.-F. Chen, J. Kiesel, H. Wachsmuth, B. Stein, Unit segmentation of argumentative texts, in: Proceedings of the 4th Workshop on Argument Mining, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 118–128. URL: https://www.aclweb.org/anthology/W17-5115. doi:10.18653/v1/W17-5115.

[11] R. Levy, S. Gretz, B. Sznajder, S. Hummel, R. Aharonov, N. Slonim, Unsupervised corpus–wide claim detection, in: Proceedings of the 4th Workshop on Argument Mining, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 79–84. URL: https://www.aclweb.org/anthology/W17-5110. doi:10.18653/v1/W17-5110.

[12] S. Eger, J. Daxenberger, I. Gurevych, Neural end-to-end learning for computational argumentation mining, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 11–22. URL: https://www.aclweb.org/anthology/P17-1002. doi:10.18653/v1/P17-1002.

[13] J. Huck, Development of a Search Engine to Answer Comparative Queries, in: Notebook for the Touch é Lab on Argument Retrieval at CLEF 2020, volume 2696 of *CEUR Workshop Proceedings*, 2020. URL: http://ceur-ws.org/Vol-2696/.

[14] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, TIRA Integrated Research Architecture, in: N. Ferro, C. Peters (Eds.), Information Retrieval Evaluation in a Changing World, The Information Retrieval Series, Springer, Berlin Heidelberg New York, 2019. doi:10.1007/978-3-030-22948-1_5.

[15] A. Bondarenko, M. Fröbe, M. Beloucif, L. Gienapp, Y. Ajjour, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, M. Potthast, M. Hagen, Overview of Touché 2020: Argument Retrieval, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), Working Notes Papers of the CLEF 2020 Evaluation Labs, volume 2696 of *CEUR Workshop Proceedings*, 2020. URL: http://ceur-ws.org/Vol-2696/.