Similar but Different: Simple Re-ranking Approaches for Argument Retrieval

Notebook of Team Hit-Girl for the Touché Lab on Argument Retrieval at CLEF 2022

Jerome Würf

[1]*Leipzig University, Augustusplatz 10, 04109 Leipzig, Germany*

Abstract

This work examines simple re-ranking approaches using the preprocessed args.me corpus to contribute to the Touché 2022 Argument Retrieval Task. The proposed retrieval system relies on an initial retrieval using a semantic search on a sentence level and takes advantage of simple heuristics. Our re-ranking approaches incorporate *maximal marginal relevance*, *word mover's distance*, and a novel approach based on a fuzzy matching on part of speech tags that we call *structural distance*. Further, we explore the applicability of a graph-based re-ranking approach. The results show that the proposed re-ranking approaches could beat our baseline. For relevance, our re-ranking using *structural distance* performs best, while for quality, the one using the *word mover's distance* achieves the highest score.

Keywords

information retrieval, argument retrieval, semantic search, re-ranking, Touché 2022

1. Introduction

The waves of protests in response to the pandemic restrictions of last winter seem to highlight a problem in the current culture of discussion. Despite an increased exposure to facts on controversial topics through our daily lives, we fail to present the gained knowledge to enable debates and to support individuals' opinion formation. Regarding COVID-19, it has been shown that people exposed to misinformation, biased media, and conspiracy have lower trust in democratic institutions [1]. This situation makes it urgent for societies to confront misinformed individuals with reasonable arguments. Besides COVID-19, web resources, like blogs and news sites, address many other topics with a similar, potentially harmful impact. This development motivates our research on the automatic retrieval of reasonable arguments.

This work, describes the submission of team Hit-Girl[1] for Task 1 of Touché 2022 [2]. The task asks participants to create an argument retrieval system for a given corpus to support the opinion formation on controversial societal topics. In this year's version of the first task, the requirements for the final systems differ from the previous years, as participants are asked to retrieve argumentative sentence pairs instead of whole arguments for a given topic. The sentence pair is reasonable if the retrieved sentences are topic-relevant and qualitative. The quality of arguments is defined by (1) the argumentativeness of each sentence, (2) coherence

CLEF'22: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

CEUR Workshop Proceedings (CEUR-WS.org)

[1] https://en.wikipedia.org/wiki/Hit-Girl

between the sentences, and (3) together, the sentences of the pair should form a summary of their originating arguments [2].

Our proposed system consists of three main components: indexing, initial retrieval, and re-ranking. The system's source code is publicly available[2]. Before indexing, sentences of the provided preprocessed args.me corpus [3] are transformed into vector embeddings. Sentences and vector embeddings are stored into two indices, one holds only premises,/ and the other holds only conclusions. We conduct a nearest neighbor search in the embedding space at retrieval time. Initially, we rank according to the cosine similarity between the query embedding and the embeddings in the respective index. This approach should maximize the semantic similarity between sentences, resulting in topic-relevant sentences. In the following, we will refer to this as *semantic search.* Finally, we compare multiple re-ranking approaches that aim to balance relevance and diversification of query results by assessing differences between a query and the retrieved sentences. Having outlined our initial motivation and a rough system overview of how we approach the given task, we pose the following research question:

Do simple, argument quality agnostic re-ranking approaches improve argument quality compared to an initial semantic search?

To answer our research question, we conducted experiments with three different re-ranking approaches utilizing *maximal marginal relevance* (MMR), *structural distance* (SD), and *word mover's distance* (WMD). In comparison to the baseline, two re-ranking approaches could increase argument relevance. WMD results in a better quality score than the baseline, and all re-ranking approaches show better sentence coherence than the baseline. Further, we analyze the challenges of implementing a graph-based argument re-ranking approach. Section 2 will introduce the related work. Following the related work, we describe our system and re-ranking approaches in section 3. Section 4 presents the evaluation of our experiments.

2. Related Work

This section introduces the challenge of argument retrieval and describes existing re-ranking approaches. We pick up on the shortcomings of previous studies to justify the design of our system.

2.1. Challenges in argument retrieval

Search engines for argument retrieval for controversial topics aim to quickly and comprehensively provide users with supportive and opposing arguments. Argument search denotes a relatively new field of research. It unites challenges of natural language processing and information retrieval while opening up a broad range of research opportunities for computational argumentation [4]. In contrast to relevance orientated search engines, systems for argument retrieval additionally needs to focus on:

- incorporating the quality of the arguments to check for their validity

[2] https://git.informatik.uni-leipzig.de/hit-girl/code

- providing an overview of arguments with different stances instead of a single best answer
- assessing and reflecting the connections between arguments in the final ranking

2.2. Existing methods in argument retrieval

ArgumenText [5] and *args* [4] are important pioneers offering diverse technical approaches to the outlined challenges of argument retrieval. ArgumentText [5] was one of the first systems ingesting heterogeneous Web documents, identifying arguments in topic-relevant documents and labeling the identified arguments with a "pro" or "con" stance. The identification of arguments relies on an attention-based neural network, and a stance recognition utilizes a BiLSTM model. Both models were trained on a dataset containing 49 topics with 600 sentences each, labeled as "pro", "con" or not an argument. The authors compare their system's performance to an expert-curated list of arguments within a specific online debate portal [3] and reported that on three selected topics, the retrieved arguments matched 89% the ones of the expert-curated list. Further, they pointed out that 12% of the arguments identified by their approach were not contained in the expert-curated list. ArgumentText [5] differs from our system as we are using a preprocessed dataset that does already contain arguments that are split into their constituent sentences. Further, these sentences are also already labeled by a stance. Therefore, our system only relies on initial retrieval and re-ranking approaches.

Args [4] is a prototype argument retrieval system using a novel argument search framework and a newly crawled Web-based corpus [3]. The framework incorporates a common argument model. In this model, one argument consists of a claim/conclusion, zero or more premises, and an argument's context, which provides the full text in which a specific argument occurred. In general, the framework splits into an indexing process and a retrieval process. The indexing process contains the acquisition of documents, argument mining, an assessment, and indexing. For the initial acquisition, the authors crawl the args.me [3] corpus. The crawl focuses on five different debate portals and includes 34,784 debates containing 291,440 arguments that were finally parsed into 329,791 argument units. Argument mining and parsing into the common argument model rely on Apache UIMA[4]. The final indexing is realized with Apache Lucene. In the retrieval process, the *args* prototype performs an initial retrieval for a given query, relying on an exact string match between query terms and terms in an indexed argument and conducts a ranking on relevant arguments using a BM25 model. To be more specific, a BM25F model was used to weigh the individual components of the common argument model. The authors performed a quantitative analysis using controversial topics from Wikipedia as queries. The scores were reported on the systems' coverage for logical combinations of query terms and phrase queries and the three components of the proposed common argument model: conclusions, arguments, and argument's context. Finally, the system achieved a good initial coverage ranging from 41.6%–84.6% for all query types on the conclusions and a coverage of 77.6% on phrase queries for whole arguments. The results indicate that a retrieval model with a higher weight on conclusions reaps arguments of higher relevance. Our system uses a preprocessed version of the args.me [3] corpus. To be specific, our system indexes sentences that were gained from the argument mining and the assessment step of the *args* search engine. Like *args*, our system's

[3] https://ProCon.org

[4] https://uima.apache.org

initial retrieval and re-ranking approaches will not rely on identifying argumentative structures within the indexed argument units. In contrast to *args*, we use two indices, one for conclusions and one for premises, instead of indexing whole arguments at once. Motivated by the findings of the args search engine that the conclusions should have higher weight, our system queries our conclusion index first and uses the retrieved conclusions to query the premises index. Furthermore, our system enforces a minimum amount of tokens in a retrieved conclusion compared to a query. This constraint is also motivated by the expectations of *args*' authors, "that the most relevant arguments need some space to lay out their reasoning"[4].

Previous years of Touché showed substantial improvements in retrieval performance. In the first year, multiple submissions indicated that the DirichletLM [6] retrieval model is a strong baseline for the initial retrieval of argumentative text [7]. Additionally, query expansion mechanisms were deployed to increase recall. Submissions for the second round of Touché indicated that argument-aware re-ranking approaches using fine-tuned language models improved previous years' results. Moreover, approaches focused on parameter tuning of pipelines proposed in the previous year, using existing relevance judgments [8]. Up to now, only a minority of Touché's submissions [9, 10] leveraged embeddings for an initial retrieval, which motivates us to gain a deeper understanding of this approach. Motivated by the promising results of query expansion of last year's submissions [11, 12, 13], our system mimics a query expansion by first retrieving conclusions with an initial controversial topic and then using these conclusions to query an index holding the premises. Finally, our re-ranking distinguishes us from existing ones, as we do not rely on argument-specific domain features or machine learning methods.

3. Methodological Approach

The architecture of our retrieval system (Figure 1) consists of indexing, retrieval, and a re-ranking module. The system relies on two Elasticsearch[5] indices, one for conclusions and one for premises. Initially, our system uses the preprocessed args.me[3] corpus, which holds arguments divided into their constituent premise and conclusion sentences. The sentences are transformed into vector embeddings (Section 3.1). Premises and conclusions are indexed into the respective indices with their vector embeddings. While indexing, the standard tokenization pipeline of Elasticsearch is applied to save the number of tokens in a sentence as further metadata.

The retrieval module generates an initial ranking for premise and conclusion pairs. First, it queries the conclusion for a given controversial topic. Next, each conclusion serves as a query for the premise index. The initial retrieval scores are based on the cosine similarity between the vector embeddings of a query and the indexed sentences, thus mimicking the nearest neighbor search in the embedding space. Additionally, we introduce the hard constraint that a retrieved sentence must have at least 1.75 times the number of tokens of the query. The exact value was chosen by convention. In the following, we will refer to this constraint as *token factor*. The usage of a *token factor* is motivated by previous findings suggesting that qualitative argumentative sentences are longer than non argumentative ones[4]. By convention, we retrieve 100 conclusions and 50 premises per conclusion. A primary motivation behind this

[5] https://github.com/elastic/elasticsearch

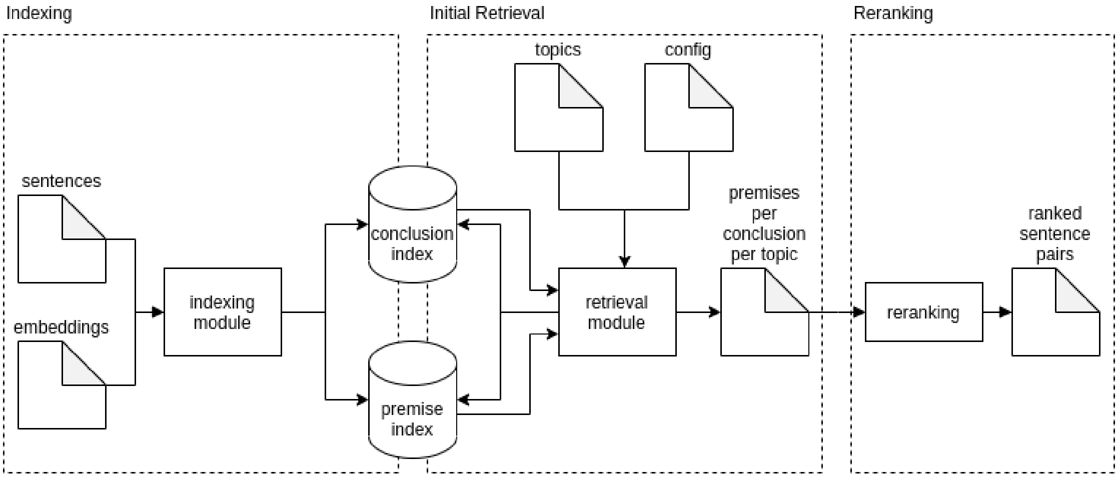


Figure 1: System architecture: Besides the preprocessing, our system splits into three parts: Indexing of sentences and embeddings, an initial retrieval on different controversial topics given a configuration, that determines the re-ranking strategy.

two-step retrieval is an expected increase in the premises recall, as we query the premise index multiple times using different conclusions.

Finally, the re-ranking module scores conclusions and premises separately using three different methods, which will be explained in section 3.2. In general, these methods should improve the ranking with respect to the argumentative quality of the retrieved sentences by calculating new ranking scores between the query and initially ranked sentences. Lastly, our system generates a text file in the "standard" TREC format. When writing the output file, we enforce on a topic level that there are no duplicates in the retrieved premises, and premises must match the stance of a conclusion.

3.1. Preprocessing

The organizers of the shared task provide preprocessed args.me [3] corpus that contains the constituent sentences of each argument of the original args.me corpus. Further, it contains context meta-data for each argument and the stance for each sentence. Initially, we transform the provided preprocessed corpus into a structured parquet file. One row of this flat file corresponds to one sentence. One row holds information about the argument ID, sentence number[6], stance towards a topic, sentence text, and the sentence type, either conclusion or premise. The flat file contains 6,123,792 sentences that split into 338,595 conclusions and 5,785,197 premises. The original argument model of args.me [3] associates one conclusion with many premises, explaining the difference in the cardinality between conclusions and premises. Our approach breaks this association and combines the premises and conclusions of different arguments. We deduplicate the sentences using an exact string match. For the conclusions, we count 328,474

[6] The sentence number presents the sentence's index in the array of premises of an argument within the preprocessed args.me

duplicates with 54,512 unique ones, which result, together with the non-duplicated ones, in a total of 64,633 conclusions to index. For the premises, we count 770,876 duplicates with 273,593 unique duplicates, which results in the non-duplicated ones in 5,626,509 premises to index. The high number of duplicates of conclusions arises from the parsing of debate platforms. Conclusions are often simply the headline of a post on a controversial topic, and a single post contains multiple arguments. The duplicated premises arise from direct citations between different posts. As a final preprocessing step, each sentence is encoded into a vector embedding via an out-of-the-box MiniLM [14] language model[7] utilizing the sentence transformers library [15].

3.2. Re-ranking approaches

To improve the argument quality of our initial retrieval, we examine three different re-ranking approaches using existing implementations. Our re-ranking approaches do not rely on argument-specific sentence features. Due to the two-step retrieval approach of our system, re-ranking scores of conclusions and premises are calculated separately. First, we re-rank the conclusions, then each set of premises is retrieved for a conclusion. Each approach combines the respective re-ranking score with the initial ranking score using a weighted sum (Section 3.2.1). We expect that this general approach improves the argumentative quality of sentences by ensuring that the top results differ from the original query. Furthermore, we explore the challenges of a graph-based argument relevance for re-ranking.

Maximal Marginal Relevance

The first sentence pairs of the results of our initial re-ranking were very similar, only differing in single words. Motivated by this observation we implement the *maximal marginal relevance* (MMR) [16]. The MMR linearly combines query relevance and the information novelty of a document within a ranking. The factor of information novelty ensures the assessed score of a document incorporates the dissimilarity towards the previously chosen ones. The tradeoff between query relevance and information novelty is controlled by a parameter λ. For our experiments, we assess different λ values. Our system calculates an MMR score for each sentence in the set of conclusions and each sentence in the individual sets of premises separately. The MMR for the conclusions calculates the query relevance between a specific conclusion and the given controversial topic and the information novelty of a specific conclusion within the set of already re-ranked ones. This approach is also conducted for every premise of the individual premise sets, where the respective conclusion serves as a query.

Structural distance

As a second re-ranking approach, we propose a re-ranking based on the *structural distance* (SD) between query and retrieved sentences. The SD should impose a penalty on retrieved sentences that merely rephrase the search query by synonyms, thus boosting the scores of sentences that have a different structure than the query. We define the structure of a sentence as a list

[7] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

of part-of-speech tags[8] generated by pre-trained pipeline[9] using the spaCy NLP library[10]. The calculations for SD (Equation 1) are closely related to the Jaro similarity. Using the part-of-speech tags of a query q and a retrieved sentence s, we calculate the Jaro similarity $sim(q, s)$ on a tag instead of a character level. The standard Jaro similarity uses the length of each string, the number of matching characters, and the number of transposed characters between both in a specific interval. We adapt this to the total number of part-of-speech tags of query $|q|$ and sentence $|s|$, the number of matching tags m, and the number of transposed tags t between query and a sentence. A matching or transposed tag within q and s counts towards m or t if it is within the window of $\left\lfloor \frac{\max(|q_{pos}|,|s_{pos}|)}{2} \right\rfloor - 1$. This approach allows for fuzzy matching based on the structure. For the calculation of the Jaro similarity, we use the popular *text-distance* package[11] and pass at the method invocation of the Jaro similarity two lists of part-of-speech tags instead of two strings. Finally, we convert the gained similarity into a distance score by subtracting it from 1, obtaining SD.

$$\begin{aligned} SD(q, s) &= 1 - sim(q_{pos}, s_{pos}) \\ \text{sim}(q_{pos}, s_{pos}) &= \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|q_{pos}|} + \frac{m}{|s_{pos}|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases} \end{aligned} \tag{1}$$

Word mover's distance

In contrast to the other two previous re-ranking approaches that examine whole sentence strings, the *word mover's distance*, proposed by Kusner et al.[17], considers the similarity of single words within two sentences to each other. For each word pair, the earth mover's distance of the corresponding words is calculated using their Word2Vec [18] embeddings. This process is formulated as a combinatorial problem to retrieve the word pairs leading to a minimal cumulative sum of distances of all constructed word pairs. Hence, it accounts for sentences with no words in common but similar meanings due to synonymy [17]. Our re-ranking leverages this behavior to rank sentences different from the query higher to provide a more diverse set of argumentative sentences. Our implementation uses the wmd-relax package[12] as it provides an off-the-shelf spaCy hook that uses the same pretrained pipeline as in SD. Using this hook, allows for an easy integration into our re-ranking pipeline.

Graph based re-ranking

Wachsmuth et al.[19] have proposed a graph-based approach to measure relevance based on structural connections between argument units. Their hypothesis states that the content of arguments does not determine their relevance. The reasoning behind this hypothesis is the subjectivity of the content of an argument. Their proposed approach infers argument relevance from the number of arguments whose conclusions serve as a premise for other arguments.

[8] https://universaldependencies.org/u/pos/

[9] https://github.com/explosion/spacy-models/releases/tag/en_core_web_md-3.3.0

[10] https://spacy.io/

[11] https://github.com/life4/textdistance

[12] https://github.com/src-d/wmd-relax

Further, the approach incorporates the intrinsic relevance of those arguments in a recursive fashion. A recursive analysis of links between argument units allows for an objective assessment of argument relevance, as no human judgment is needed. The authors adopt vital components of the PageRank algorithm [20]. They use a framework of argument graphs, in which the arguments represent nodes. Arguments are split into premise and conclusion as argument units. Reusing a conclusion as a premise in another argument determines an edge between two argument nodes. An edge is constructed based on an interpretation function. The authors use an exact string match as an interpretation function.

Using our nested structure (multiple conclusions per topic and multiple premises per conclusion) provided by the initial retrieval step, we model one argument graph for each topic using the *networkX* [21] graph processing library. For edge interpretation, we reuse the vector embeddings of the initial retrieval and calculate the cosine similarity between each premise and all the other conclusions. If an interpretation threshold of .99 is surpassed, we would create an edge. Analyzing the threshold surpassing similarities over all topic-based argument graphs, we observe a high skew within the similarities (Figure 2, right). The skew can be attributed to the initial retrieval that is also based on the cosine similarity. Regarding the connectivity between arguments, the skewed distribution of cosine similarities leads to few highly connected argument nodes and a majority of nodes with only a single connection to another argument (Appendix 3a). In our system's setup, an application of a PageRank for arguments would not lead to any meaningful re-ranking scores.

Furthermore, we investigate the WMD for graph construction. Using the WMD instead of the cosine similarity should better assess the semantic differences between two argument units. We transform the WMD into a similarity to use it as an interpretation function. We call the transformed measurement *word mover's similarity* (WMS). WMS is gained by the following transformation $wms(s_1, s_2) = \frac{1}{1+wmd(s_1,s_2)}$. We assess an initial interpretation threshold of 0.2 that must be surpassed to draw an edge between two arguments. The similarity distribution over all topics differs tremendously from the distribution of cosine similarities (Figure 2, left). Nevertheless, similar to the argument graphs generated using cosine similarity, the node degree distribution is also skewed (Appendix 4b).

Next, we examined the total amount of edges for every topic for both interpretation functions (Appendix 3b and 4b). Due to the lower interpretation threshold of the WMS compared to the argument graph construction with the cosine similarity, the number of total edges in the argument graphs is higher. Increasing the threshold would lead to topics for which no WMS would surpass the threshold, thus not generating an argument graph. To alleviate the problem, an individual topic threshold must be tuned, questioning the general applicability of the re-ranking approach.

Finally, some edges could attribute the wrong arguments due to our initial deduplication of the provided corpus. The duplicated premises in the provided corpus originated from arguments citing each other within the crawled debate platforms. Our sentence level deduplication is based on exact string matches, and only the first occurrence of a sentence is kept, while the others are discarded. There, we could not enforce that a particular sentence is linked to the argument ID of the original argument, where it was written the first time. Due to the challenges outlined in this section, we did not further investigate a re-ranking based on argument graphs.

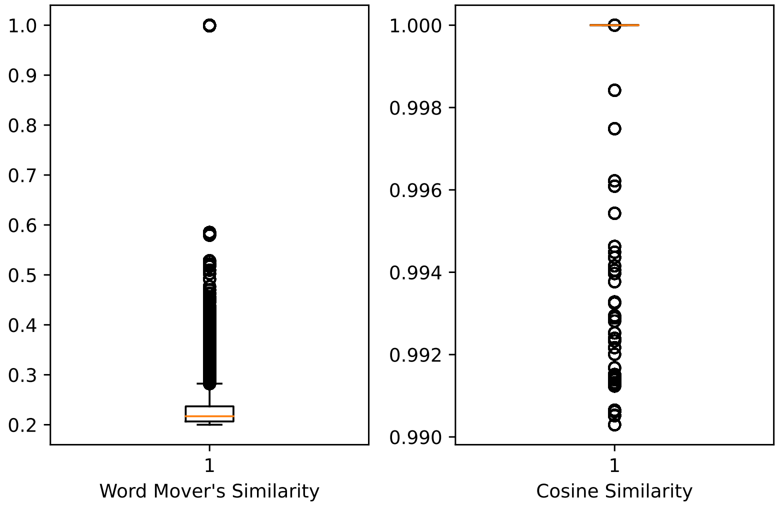


Figure 2: Comparison of the distribution of similarities between Word Mover's Similarity and Cosine Similarity.Word Mover's Similarity is gained by rescaling the Word Mover's Distance. Similarity values were gained by combining the similarities of the constructed argument graphs overall provided sample topics. Argument graph construction using word mover's similarity used an interpretation threshold of 0.2 to create an edge between two arguments, and the argument graph construction using cosine similarity used a threshold of .99.

3.2.1. Final Scoring

Our system calculates the final re-ranking scores for conclusions and premises separately. For WMD and SD, we assess the final score S of a premise or conclusion as a weighted sum between the initial cosine similarity I and the respective re-ranking score R (Equation 2). MMR does not need a weighted sum, as the MMR itself includes the initial re-ranking score information. Like λ of the MMR, μ controls the tradeoff between sentence relevance and difference to the query. A higher μ emphasizes the initial ranking scores and penalizes the re-ranking score. S denotes the final score between a query sentence q and an indexed sentence d. We scale both I and R to the interval of $[0, 1]$. For each SD and WMD, we examined different parameters of μ, relying on our qualitative assessment of the generated rankings. For our final evaluations, we set μ to the values of 0.9 for conclusions and 0.75 for premises. These parameter configurations were determined by a heuristic assessment of the relevance and quality of the generated rankings.

$$S_{(q,d)} = \mu * I(q, d) + (1 - \mu) * R(q, d) \tag{2}$$

4. Evaluation

We performed four runs on the TIRA platform [22] to ensure the reproducibility of our results. Runs are named after the planet Jupiter and the first three Galilean moons. The evaluation foots on the judgments that the task organizers provided. The reported scores adhere to the

Approach (TIRA tag)	Relevance	Quality	Coherence
Baseline (Jupiter)	0.560	0.725	0.330
SD (Io)	**0.588**	0.719	0.365
MMR (Europa)	0.546	0.721	0.349
WMD (Ganymede)	0.583	**0.776**	**0.377**

Table 1
Resulting mean nDCG@5 for relevance, quality and coherence-based evaluation over 50 topics. Two of our re-ranking approaches could beat the baseline without any re-ranking on relevance. WMD achieves the highest quality score. All runs use a *token factor* of 1.75 for the initial retrieval. The the runs of SD and WMD use $\mu = 0.9$ for the re-ranking of conclusions and $\mu = 0.75$ for the re-ranking of premises. MMR uses a $\lambda = 0.75$.

recommendation of calculating nDCG@5. Table 1 shows our relevance, quality, and sentence pair coherence results. To measure the effectiveness of our implemented re-ranking methods, we include a baseline (Jupiter) that relies on the initial retrieval based on the cosine similarities generated by Elasticsearch. Two re-ranking approaches, SD and WMD, could beat the baseline with a mean nDCG@5 of 0.588 and 0.583 for relevance. The run using SD ranks 12th for relevance among all submitted runs of the shared task. SD could not outperform the baseline regarding the quality. WMD showed the best quality measurement, ranking at 8th place among all participating runs. For relevance and quality, the MMR (Europa) experiment did worse in comparison to the baseline but slightly better than SD on quality. Unsurprisingly, none of our runs could achieve high scores regarding the coherence between two sentences, as our retrieval system does not optimize for this criteria.

5. Conclusion

We examined whether re-ranking approaches that do not make inferences about argument quality can improve rankings generated by an initial semantic search. In our theory, the initial search maximizes topic relevance, and the argument agnostic re-rankings increase variety, potentially ranking more qualitative sentence pairs of premise and conclusion higher. We have implemented an argument retrieval system using word embeddings for the initial ranking and three argument quality agnostic re-ranking approaches to answer our research question. The re-ranking approaches foot on the *maximal marginal relevance*, the *word mover's distance*, and a novel distance measure based on a fuzzy matching on sentence tags, which we call *structural distance*. The results show that simple re-ranking approaches could outperform our baseline without re-ranking by a small margin. Our system introduces several parameters. The initial ranking uses a *token factor*, *maximal marginal relevance* imposes λ and *structural distance* and *word mover's distance* use a weighting factor of μ. For the next iteration of Touché, when relevance and quality judgments on a sentence pair level are available, we will perform parameter fine-tuning to improve our outlined approaches in future research.

References

[1] L. Pummerer, R. Böhm, L. Lilleholt, K. Winter, I. Zettler, K. Sassenberg, Conspiracy Theories and Their Societal Effects During the COVID-19 Pandemic, Social Psychological and Personality Science 13 (2022) 49–59.

[2] 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.

[3] Y. Ajjour, H. Wachsmuth, J. Kiesel, M. Potthast, M. Hagen, B. Stein, Data Acquisition for Argument Search: The args.me Corpus, 2019, pp. 48–59.

[4] 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.

[5] 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.

[6] C. Zhai, J. Lafferty, A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval, in: ACM SIGIR Forum, volume 51, ACM New York, NY, USA, 2017, pp. 268–276.

[7] A. Bondarenko, M. Hagen, M. Potthast, H. Wachsmuth, M. Beloucif, C. Biemann, A. Panchenko, B. Stein, Touché: First Shared Task on Argument Retrieval, in: P. Castells, N. Ferro, J. Jose, J. Magalhães, M. Silva, E. Yilmaz (Eds.), Advances in Information Retrieval. 42nd European Conference on IR Research (ECIR 2020), volume 12036 of *Lecture Notes in Computer Science*, Springer, Berlin Heidelberg New York, 2020, pp. 517–523.

[8] 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.

[9] R. Agarwal, A. Koniaev, R. Schaefer, Exploring Argument Retrieval for Controversial Questions Using Retrieve and Re-rank Pipelines, in: CEUR Workshop Proceedings, 2021, pp. 2285–2291.

[10] K. Ros, C. Edwards, H. Ji, C. X. Zhai, Team Skeletor at Touché 2021: Argument Retrieval and Visualization for Controversial Questions, in: CEUR Workshop Proceedings, 2021, pp. 2441–2454.

[11] C. Akiki, M. Fröbe, M. Hagen, M. Potthast, Learning to Rank Arguments with Feature Selection, in: CEUR Workshop Proceedings, 2021, pp. 2292–2301.

[12] E. Raimondi, M. Alessio, N. Levorato, A Search Engine System for Touché Argument Retrieval Task to Answer Controversial Questions, in: CEUR Workshop Proceedings, 2021, pp. 2423–2440.

[13] A. Mailach, D. Arnold, S. Eysoldt, S. Kleine, Exploring Document Expansion for Argument Retrieval, in: CEUR Workshop Proceedings, 2021, pp. 2417–2422.

[14] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, M. Zhou, Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, Advances in Neural Information Processing Systems 33 (2020) 5776–5788.

[15] N. Reimers, I. Gurevych, N. Reimers, I. Gurevych, N. Thakur, N. Reimers, J. Daxenberger, I. Gurevych, N. Reimers, I. Gurevych, et al., Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019, pp. 671–688.

[16] J. Carbonell, J. Stewart, The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries, SIGIR Forum (ACM Special Interest Group on Information Retrieval) (1999).

[17] M. J. Kusner, Y. Sun, N. I. Kolkin, K. Q. Weinberger, From Word Embeddings to Document Distances, in: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, JMLR.org, 2015, p. 957–966.

[18] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, arXiv preprint arXiv:1301.3781 (2013).

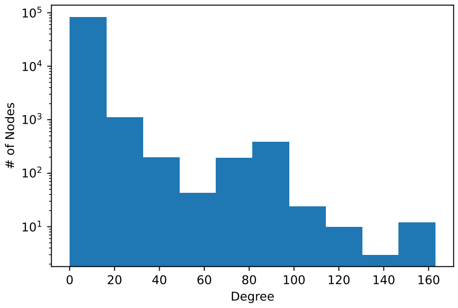
[19] H. Wachsmuth, B. Stein, Y. Ajjour, "PageRank" for Argument Relevance, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, 2017, pp. 1117–1127.

[20] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web., Technical Report 1999-66, Stanford InfoLab, 1999.

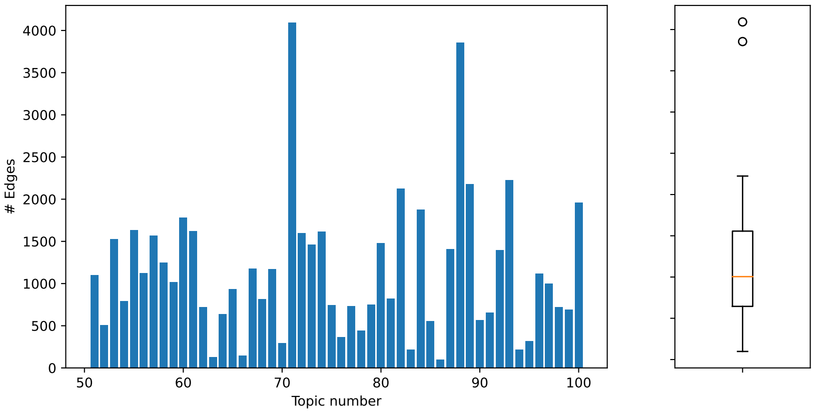
[21] A. A. Hagberg, D. A. Schult, P. J. Swart, Exploring Network Structure, Dynamics, and Function using NetworkX, in: G. Varoquaux, T. Vaught, J. Millman (Eds.), Proceedings of the 7th Python in Science Conference, Pasadena, CA USA, 2008, pp. 11 – 15.

[22] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, Tira Integrated Research Architecture, in: Information Retrieval Evaluation in a Changing World, Springer, 2019, pp. 123–160.

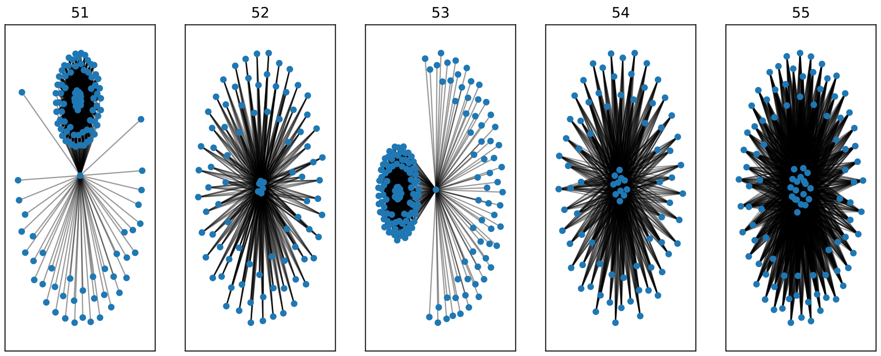
A. Argument Graphs: Edge Interpretation based on Cosine Similarity



(a) Node degree histogram over all topics.

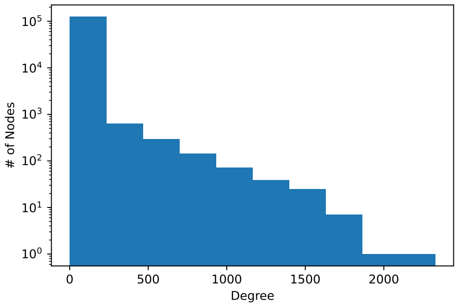


(b) Total count of edges between arguments per topic.

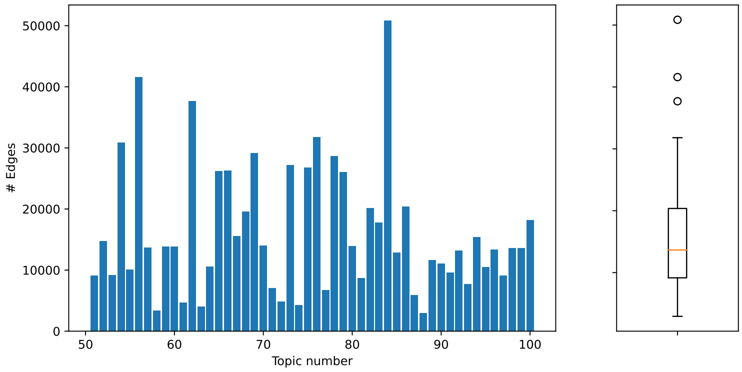


(c) Example graphs for five topics.

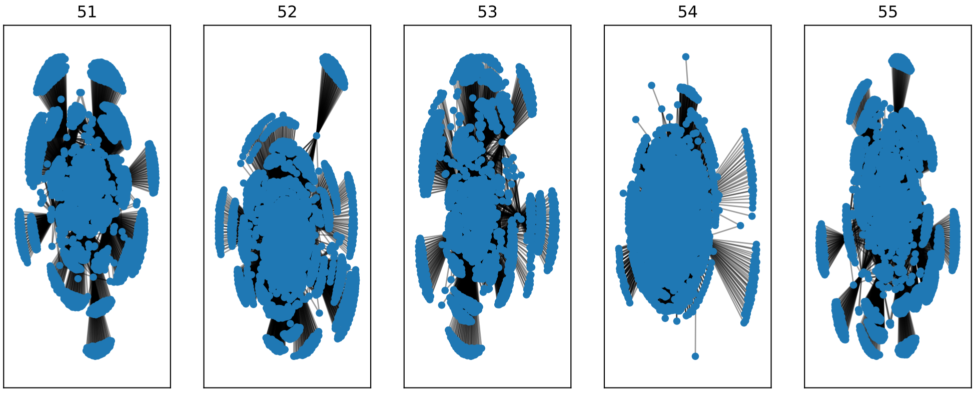
B. Argument Graphs: Edge Interpretation based on Word Mover’s Distance



(a) Node degree histogram of all topics.



(b) Total count of edges between arguments per topic.



(c) Example graphs for five topics.