The Pearl Retriever: Two-Stage Retrieval for Pairs of Argumentative Sentences

Notebook for the Touché Lab on Argument Retrieval at CLEF 2022

Sebastian Schmidt[1], Jonas Probst[1], Bianca Bartelt[1] and Alexander Hinz[1]

[1]*Leipzig University - Department of Computer Science*

Abstract

This paper describes the retrieval approach submitted by Team Pearl to the first Touché shared task at CLEF 2022 [1]. The model combines two retrieval pipelines to obtain pairs of argumentative sentences for a given query that relates to a controversial topic. The first pipeline uses a Dirichlet model to identify relevant arguments while the second applies a DPH model to retrieve relevant sentences. Both sentences and arguments are filtered using pre-calculated scores of argumentative quality and only sentences that belong to one of the remaining arguments are presented as results.

We experimented with reranking retrieved arguments using an adapted version of the ArgRank proposed by Wachsmuth et al. [2] but did not find our implementation to improve retrieval performance beyond chance effects. Furthermore, we evaluated different approaches of matching sentences to form coherent pairs and found the naive approach of choosing partners from the immediate neighbourhood of a sentence in its parent argument to outperform more sophisticated solutions.

Keywords

Argument retrieval, Controversial questions, ArgRank, Sentence matching

1. Introduction

Popular web search engines are a central access point to the vast range of available information on the Web. Despite that position, they currently do not address the challenges around retrieving argumentative texts like the assessment of argumentative quality [3]. With our submission to the third Touché lab, we aim to contribute to the lab's goal of trying to close this research gap by exploring how to identify pairs of sentences that represent arguments relevant to a user's opinion formation process on a controversial topic. A relevant argument is one that both discusses the topic at hand as well as fulfills quality criteria like logical coherence. A pair of sentences can be considered representative of such an argument if it contains a central take-away from the argument, is itself relevant to the topic, and is both argumentative and coherent. We aim to address these challenges with our retrieval model that consists of two separate retrieval pipelines and quality assessments, ArgRank-based reranking [2], and sentence matching that aims for high coherence and quality in the resulting pair.[1]

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

ss56pupo@studserv.uni-leipzig.de (S. Schmidt); gv53ozag@studserv.uni-leipzig.de (J. Probst); bianca.bartelt@studserv.uni-leipzig.de (B. Bartelt); ah25tixi@studserv.uni-leipzig.de (A. Hinz)

[1] See https://git.informatik.uni-leipzig.de/ah25tixi/informationretrieval_pearl for our repository.

2. Related Work

In the following, we refer to previous work that provided the inspiration and basis for our retrieval approach. First, references are made to argument structure, then to argument relevance, and finally to argument and sentence quality.

2.1. Argument Structure

The two central units of retrieval for the context of this paper are arguments and sentences taken from these arguments. As our approach was inspired by the framework for argument search introduced by Wachsmuth et al. [4], this paper uses the argument model put forth in their paper. This model builds on the common argument structure of conclusion and premises, with the former being the main claim of an argument and the latter sentences that discuss the argument's topic to arrive at the conclusion [5].

2.2. Argument Relevance

The retrieval of argumentative sentences is a classic example of information retrieval. According to Stein et al. [6], a retrieval task consists of responding to an information need using an information resource like a collection of documents. The goal of this task is to identify a subset of documents from the information resource that the user considers relevant to their information need. This is achieved by retrieval models which aim to rank the available documents according to the probability that they are relevant to a given information need stated in the form of a query.

While there is a wide range of general retrieval models available, no specific model was developed for the task of argument retrieval so far. Given this research gap, Potthast et al. [7] assessed the performance of four popular retrieval models in this field . Specifically, they applied each model to the task of argument retrieval on a set of 20 controversial issues, represented by a neutral and a biased query per issue, and evaluated the models' performance on the relevance of the results to the query as well as three measures of argument quality (rhetorical, logical, and dialectical quality). The results showed that the DPH [8] and Dirichlet [9] model both clearly outperform BM25[10] and TF-IDF [11], with DPH achieving the best performance on the relevance metric and Dirichlet demonstrating lower variance than DPH as well achieving a higher score on two of the quality metrics. Building on these findings, we applied both models to the retrieval of argumentative sentence pairs.

The Dirichlet model is a language model approach, a subcategory of retrieval models that develop document-specific language models and rank documents based on the probability that their respective language model generated the query [6]. This relevance calculation is derived from Bayes' theorem applied to the conditional probability for a document d_i given query q:

$$P(d_i|q) = \frac{P(q|d_i)P(d_i)}{P(q)} \tag{1}$$

Given that the probability of a query $P(q)$ is the same for all documents, it is omitted from the equation. Furthermore, the probability distribution for all documents is assumed to be uniform and $P(d_i)$ is therefore discarded as well as it wont affect the ranking between documents. This uniformity assumption can however it be dropped in favor of query-independent relevance judgements that estimate the document-specific probabilities $P(d_i)$. One way to achieve this was suggested by Wachsmuth et al. [2] with their reinterpretation of the PageRank algorithm [12] for the application of argument search. The authors distinguish between the local and global relevance of an argument and its parts, with the former assessing if its premises are relevant to the conclusion of the argument and the latter evaluating the contribution of an argument as a whole to resolving a debate [13][14]. The "ArgRank" estimates the global relevance of an argument based on other arguments that use its conclusion as a premise, thereby assuming that the relevance of a conclusion to a given discussion is reflected by how many other arguments in that discussion refer to it. Similar to PageRank, the ArgRank is calculated as the weighted sum of a ground relevance (left term in equation 2) and a recursive relevance (right term) that rewards both a high rank as well as a low number of outgoing links of referring documents.

$$\hat{p}(c_i) = (1-\alpha) * \frac{p(d) * |D|}{|A|} + \alpha * \sum_{j=1} \frac{\hat{p}(c_j)}{|P_j|} \tag{2}$$

- $\hat{p}(c_i)$ ArgRank of argument i
- α Weighting factor between 0 and 1
- $p(d)$ PageRank of the web page that c_i is stated on
- $|D|$ Number of web pages in the collection
- $|A|$ Number of arguments on all web pages combined
- $\hat{p}(c_j)$ ArgRank of an argument j that uses c_i as a premise
- $|P_j|$ Number of premises of argument j

The DPH model, finally, belongs to the family of probabilistic models that treat relevance as a binary event (1=relevant and 0=irrelevant) and try to estimate the probability of a document d_i being relevant to the query q [6]. The model's central assumption can be summarized as follows: If a given term t_j of the query q is both relatively rare in the overall document collection D and relatively common in a specific document d_i, then t_j has a high information content for d_i. While this relation generally follows a hypergeometric distribution, it can be reduced to the following binomial distribution for large document collections and comparably short documents where $|d_i|$ is the number of terms in d_i, f_{ij} is the number of occurrences of t_j in d_i and $P(t_j)$ is its frequency in D:

$$B(|d_i|, f_{ij}, P(t_j)) = \binom{|d_i|}{f_{ij}} * P(t_j)^{f_{ij}} * (1 - P(t_j))^{|d_i| - f_{ij}} \tag{3}$$

Hence, the model calculates the probability of observing f_{ij} occurrences of t_j in d_i given (i) the probability $P(t_j)$ of observing the term in the underlying population and (ii) the number of terms $|d_i|$ of the document d_i. In order to turn an unlikely observation into a high information value, the resulting probability is transformed using the binary logarithm multiplied by minus one:

$$Inf(f_{if}||d_i) = -\log_2[B(|d_i|, f_{ij}, P(t_j))] \tag{4}$$

This equation for the information value is then used to estimate the relevance of a document d_i based on the terms t_j of the query q.

2.3. Argument and Sentence Quality

Argument Quality has been thoroughly discussed by Wachsmuth et al., who identified three main dimensions [15]. First, logical quality assesses if the argument is well structured and correctly builds on premises to form its conclusion. Second, rhetorical quality represents how well-written and persuasive the argument is in the given context. And finally, dialectical quality evaluates how important the argument is for resolving the current discussion.

Based on these quality dimensions, Gienapp et al, introduced the Webis Argument Quality Corpus 2020 [16], which contains 1,610 text spans in total, of which 339 were annotated as non-arguments while the remaining 1,271 arguments were annotated in the given quality dimensions on a scale between -4 (spam) and 3 (high quality). In addition to that, these arguments also received a combined quality score.

The scores were calculated using pairwise annotations by crowd workers, which has shown to yield results of higher quality in comparison with traditional rating methods in which the argument quality is assessed directly.

For the Touché Task 2021, Team Yeagerists [17] used this dataset to train a quality estimation model based on BERT [18] to refine their argument ranking function. Their best model achieves an R^2-Score of 0.7439 and a MSE of 0.7280 on the test set.

The model architecture is based on the base model from Gretz et al. [19], who compare different versions of BERT models to predict sentence quality on the IBM-Rank-30k dataset, which was also introduced in [19]. It consists of over 30,000 sentences which were evaluated for quality on a scale between 0 and 1. Their base model is the pre-trained BERT model without fine-tuning to which a fully-connected hidden layer with 100 neurons is added, feeding their outputs to a sigmoid activation layer to produce a single output. The base model achieves a Pearson correlation of 0.48 and a Spearman Correlation of 0.43 on the test set. This is a significant improvement over the method based on GloVe embeddings and even more so over the simple baseline that only considers argument length.

In our work, we used both models for predicting argument and sentence quality, respectively, and implemented them using the Huggingface library and Pytorch, leaving the model architectures unchanged. For the sentence model, we achieved Pearson correlation of 0.489 and a Spearman Correlation of 0.436, replicating the quality scores of Gretz et al. [19]. For the argument model, we achieved an R2-score of 0.7, which is worse than the 0.74 achieved by Team Yeagerists [17]. Since we replicate their architecture exactly, this difference can likely be attributed to the random split in training, validation and test set. The results are still comparable, so the model is used as is.

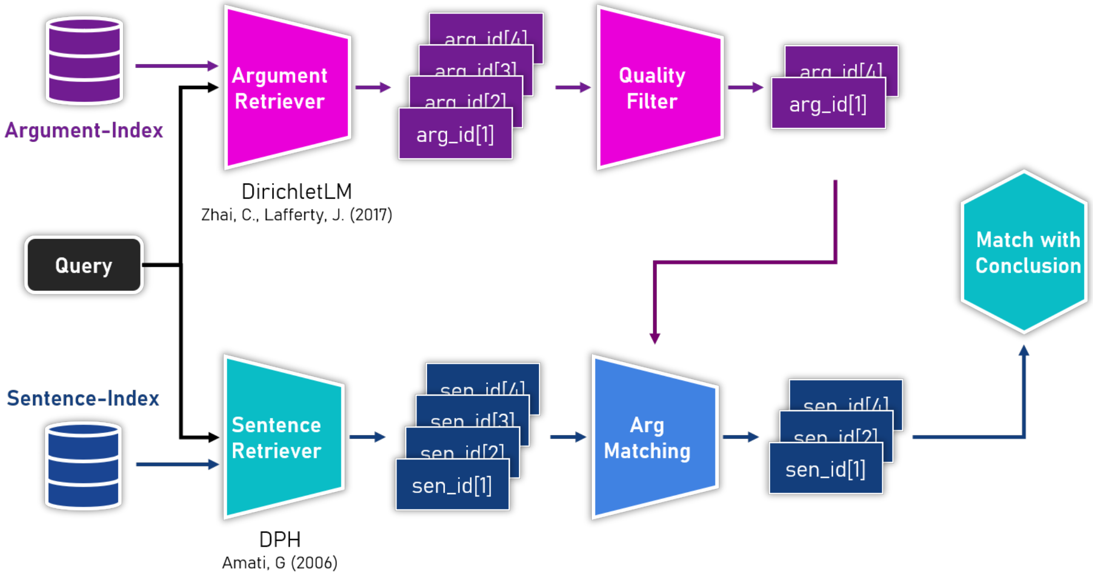

Figure 1: An overview of the vertical prototype, consisting of the two parallel pipelines for argument and sentence retrieval.

3. The Retrieval Model

Our retrieval model was developed in iterative steps that are described below. After preprocessing the dataset, we developed a vertical prototype to explore the task and identify potential for improvement. These learnings were then used to develop the refined prototype.

3.1. Dataset

The underlying dataset is taken from the args.me corpus [20], collected for the development of argument search engines. It consists of roughly 360,000 arguments crawled from debating platforms such as idebate.org, debatepedia.org, debatewise.org and debate.org. In addition to the argument text grouped into conclusion and premises, each argument has a set of meta data including the title of the discussion it was stated in and the stance of the argument to that discussion. In order to perform retrieval on individual sentences, the premises and conclusions of the args.me corpus were provided as an additional document collection.

After an initial exploration of the argument and sentence collection, we implemented an idea developed by Gienapp[21] for Touché 2021 to clean the dataset of non-argumentative documents. In their approach, the authors trained a Support Vector Machine to predict if a given text is argumentative based on the "Is Argument?"-label in the Webis Argument Quality Corpus 2020. We used the same approach and achieved the same F1-score of 0.88, using 10-fold cross-validation. The 63,019 arguments that were removed from the corpus have an average quality of -1.76 according to our argument quality prediction model. This shows a strong agreement between the non-argument-label given in the Webis Argument Quality Corpus 2020 and the quality scores our model predicts.

3.2. Vertical Prototype

The first step in constructing the retrieval approach was to develop a vertical prototype in order to both explore our initial ideas as well as create an evaluation baseline against which to compare the effect of any subsequent changes to the approach. This vertical prototype consists of a combination of two parallel retrieval pipelines, visualized in Figure 1. The first pipeline uses Dirichlet to retrieve relevant arguments, while the second directly retrieves sentences with DPH. Both retrieval models are implemented using PyTerrier, the Python API for Terrier [22]. After obtaining a ranked set of retrieved arguments with Dirichlet, these arguments are filtered based on the pre-computed quality scores from the argument quality estimation model described in chapter 2.3. The remaining arguments are then used to filter the set of sentences retrieved by the DPH model. This is done to ensure that the remaining sentences are not only deemed relevant by the DPH model, but are also taken exclusively from qualitatively good arguments that are relevant to the query. In the last step, the retrieved sentences are matched with the conclusion of their respective argument to create the final sentence pairs. These pairs are then ranked according to the relevance score of the sentence retriever.

The performance of this initial approach was evaluated on a set of 35 queries taken from the controversial topics used in the Touché Task 2021 [23]. For each of these queries, the ten most relevant sentence pairs were obtained and evaluated across three metrics.[2] In addition to providing a baseline for the evaluation of changes to the retrieval process, this also yielded two important insights on issues with the existing approach.

The first discovery of the baseline evaluation was that the reason the DPH model was chosen for sentence retrieval, namely its strong focus on documents that contain relatively rare terms of the query, is also one of its weaknesses. On the one hand, the model is susceptible to homonyms and terms being used in a different context than that of the query. For instance, the query "Should Insider Trading Be Allowed?" yielded sentences debating whether or not the attacks on September 11, 2001 were an "inside job". This behavior could be one explanation of the relatively high score variance that Potthast et al. found for DPH in comparison with Dirichlet-based retrieval [7]. On the other hand, the model assigns a high relevance to sentences even if the terms are only used as part of a URL or other types of sources. This behavior negatively affects the task-specific retrieval performance as some highly ranked results were not argumentative premises but a list of sources for the respective argument.

The second discovery was that matching all retrieved sentences with the conclusion of their parent argument led to large quality variations in the results. This circumstance is a consequence of the way that the args.me corpus was created. As the majority of arguments crawled from the debate platforms did not have a dedicated "conclusion"-field, this missing data was imputed with the title of the discussion that the argument was stated in. Given that most arguments on these platforms take a stance (PRO/CON) to a discussion title, these imputed conclusions occasionally reflect a position opposite to that of the argument they belong to. Hence, in some instances, using the conclusions as partners for the retrieved sentences resulted in incoherent sentence pairs and a failure to correctly represent the parent argument.

[2] A more detailed description of the evaluation process is provided in chapter 4.

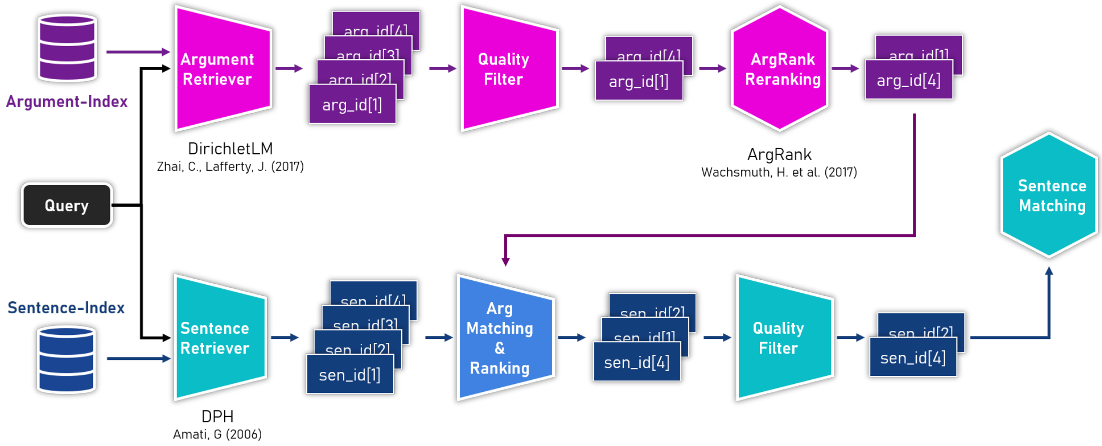

Figure 2: The final retrieval approach of the refined prototype.

3.3. Refined Prototype

The features of the refined prototype shown in Figure 2 were largely chosen based on the insights gained in the evaluation of the vertical prototype. However, some initial changes were made before addressing the issues outlined above. Firstly, we introduced a quality filter for sentences, which was trained on the IBM-Rank-30k dataset [19], as described in chapter 2.3. Secondly, we added a query expansion to address the problem of term mismatch by expanding the original query with new, related terms. Specifically, we applied the Bo1 query expansion as implemented in the PyTerrier platform. Bo1 is a "Divergence From Randomness"-weighting model based on the Bose-Einstein statistics [24] and research on query expansion has shown it to be effective in finding additional terms for a search query [25][26]. The Bo1 model achieves this goal by extracting terms from the top ranked documents obtained for the original query, weighting them based on their informativeness and adding the highest weighted terms to the original query.

After adding the sentence quality filter and query expansion, the identified weaknesses of the vertical prototype were addressed. The effect of these changes was evaluated on a reduced set of ten topics from the Touché Task 2021 [23], containing the five best and five worst performing topics for the vertical prototype. For each of these topics, the ten most relevant sentence pairs were used to calculate the nDCG@10, with the vertical prototype achieving a score of 0.4977.

DPH's Weaknesses

The first measure aimed to reduce DPH's susceptibility to query terms used as part of URLs inside arguments. Therefore, the argument as well as the sentences collection were preprocessed by replacing all instances of URLs with placeholders in the form of "[URL]" and calculating a new index. As a consequence, only those sentences that contain query terms in the "regular" text body are rewarded by DPH. Afterwards, to reduce the bias that is introduced by terms being used in different contexts than that of the query, the calculation of the final relevance score was adapted. While the relevance scoring of the vertical prototype was based entirely on the DPH model, the refined prototype also incorporates the relevance scores of the sentences' parent arguments as obtained by the Dirichlet model. The idea behind this solution was to use the lower variance of Dirichlet to “stabilize” the retrieval results and reward sentences from arguments with a high estimated relevance to the given query. After the adaptation, the retrieval results were initially ranked by their argument relevance first and sentence relevance second (i.e. within the same argument). This solution, however, comes with its own drawback as the model returns all retrieved sentences from the most relevant argument before moving on to those from the second most relevant argument and so forth. As a consequence, a sentence with the highest relevance according to the DPH model will only be returned as the most relevant result if it is also part of the argument with the highest relevance according to the Dirichlet model.

Therefore, as the final change to the refined prototype, the total score was calculated based on a weighted sum between the argument and the sentence score. The weighting factor was calculated using the generalized reduced gradient method as implemented by the Solver add-in for Microsoft Excel to find the weighting that would have resulted in an optimal ranking of existing retrieval results. As this step was applied at the end of model development, the optimal weighting factors were identified for the two best performing approaches: The "blocklist"-model and the "ArgRank"-model that uses all features of the former and re-ranks results using ArgRank. The identified weighting factor for sentence relevance was 0.38 in case of the ArgRank-model and 0.51 for the blocklist-model with the argument relevance being assigned the remaining weight. A more detailed description of the two approaches can be found in the corresponding sections below. The optimally weighted scores resulted in nDCG-values of 0.7332 for the blocklist-model and 0.7352 for the ArgRank-model.

Sentence Matching

As the pairing of a sentence with its conclusion proved ineffective, we experimented with three different approaches to identify a partner for a given sentence. The first approach consisted of using Next Sentence Prediction (NSP) [18] based on BERT-encodings. In more detail, the sentences that remain after filtering by retrieved arguments and sentence quality are used as a set of pairing candidates R. The sentences in this set are then processed in order of estimated relevance with the most relevant sentence s_i being combined with all other available sentences. After identifying the partner s_j that is most likely to follow s_i, both sentences are removed from R to avoid redundant results and the process continues with the next most relevant sentence. This matching procedure did not improve retrieval performance over that of the vertical prototype as it resulted in a relatively low nDCG of 0.4255. In our manual evaluation we found that the NSP tends to value sentences higher which contain the same or similar words as the first sentence. In many cases, this results in the second sentence only repeating the content of the first sentence, and thereby not providing new information.

In order to increase diversity in the sentence pairs, the second approach was inspired by the Maximal Marginal Relevance (MMR) that Carbonell and Goldstein [27] introduced as a result set diversification method. Instead of selecting retrieval results by weighting between relevance to the query and similarity to the already existing retrieval set, our approach forms sentence pairs by weighting between NSP-score and cosine similarity of the BERT-encodings. This approach also works iteratively over the set of pairing candidates R in order of relevance and determines the best partner for a sentence s_i by computing a score for each possible candidate s_j in R.

$$\max_{s_j \epsilon R \setminus \{s_i\}} [\lambda * sim_1(s_i, s_j) - (1 - \lambda) * sim_2(s_i, s_j)] \tag{5}$$

We implemented this approach from scratch as follows: The score is calculated using the linear combination of sim_1 (normalized NSP) and sim_2 (cosine similarity between s_i and s_j). Again, the sentence s_j with the highest score is designated as the partner for s_i and both are removed from the set of pairing candidates R. When sim_1 and sim_2 were equally weighted (λ = 0.5), we obtained a very low nDCG of 0.2801. Setting λ equal to 1 returns the Next Sentence Prediction mentioned above. Given these results, we concluded that punishing cosine similarity negatively affects retrieval performance and did not explore other levels for λ.

The final method for sentence matching we explored, was the naive approach of choosing a partner from a sentence's "neighbourhood" in its parent argument. For a retrieved sentence s_i, its partner is chosen to be either the preceding sentence s_{i-1} or the following one s_{i+1}. The choice depends on which of the two candidates produces the higher quality score when matched with s_i. We calculated the quality between adjacent sentences using the same model that we had previously used for individual sentence quality. While the scores and pairs are pre-calculated in this approach, the set of retrieved sentences R is still processed iteratively and any used sentence removed from it to avoid duplicates. Among the three approaches applied to sentence matching, this one clearly outperformed the other two with a nDCG of 0.6593 and was thus chosen for our refined prototype.

Blocklist

A comparably small change that proved effective was the inclusion of a blocklist for certain sentence parts. During multiple manual evaluation sessions, we noticed a recurring pattern of sentences in the retrieval results that either did not contain arguments or even cited positions of an opposing stance. Those sentences often contained specific statements like "my opponent claims...", "PRO claims.." or "I accept this debate", which are commonly used on debate platforms. The addition of a blocklist that filters out sentences containing these phrases led to an nDCG of 0.6914 and thus showed clear improvement over the previous value of 0.6593 for neighbour matching.

Reranking Using ArgRank

As described in chapter 2.2, the probability calculation of the Dirichlet retrieval model can be extended to also include a query-independent document probability in the form of an ArgRank. The first step towards calculating this probability is to construct a directed argument graph with edges $e = (d_j, d_i)$ denoting that argument j uses the conclusion c_i of argument i as one of its premises p_{jk} (The subscript k refers to the position of the premise in j). In order to find these edges, reuses of c_i need to be identified by searching for semantically equivalent premises in the sentence collection. While determining semantic equivalence continues to be a difficult challenge [2], transformers have proven very capable at encoding semantic information of text passages. The specific model we chose for this task is the MPNet proposed by Song et al. [28]. Both the conclusion c_i and all premises p_{jk} from a set of candidates P_c are encoded using this sentence transformer. The semantic similarity between c_i and every $p_{jk} \in P_c$ is then calculated as the cosine similarity between the encodings.

If the cosine similarity between p_{jk} and c_i is above a threshold of 0.7, the edge (d_j, d_i) is added to the argument graph together with the specific similarity score $sim(c_i, p_{jk})$ and the number of premises $|P_j|$. The search space for the set of candidate premises P_c is restricted to those arguments that both were stated in the same discussion as well as have the same stance towards the discussion's topic as argument i. This more conservative approach was chosen to increase the probability that a semantically similar sentence constitutes a reuse of c_i as it is stated in support of an "allied" argument in the same discussion. Furthermore, given that discussion titles were used to impute missing conclusions in the collection, the search for edges of the graph was only conducted for arguments with a conclusion that is different from the corresponding discussion title.

After constructing the argument graph, the next step consisted in calculating the ArgRank. Here, we again made some alterations to the approach suggested by Wachsmuth et al. [2]. First, as all arguments in the collection were obtained from debate platforms, the relevance of their parent documents (i.e., the web pages they were taken from) is assumed to be equal for all arguments. Hence, the term reflecting the ground relevance is normalized using $|A|$, the number of arguments in the collection. Second, we experimented with using the cosine similarity of an edge in the argument graph as a weighting factor of the recursive relevance. This was done to evaluate how punishing lower semantic similarity scores would affect retrieval performance. These changes lead to the following, adapted version of the ArgRank from equation 2 where $sim(c_i, p_{jk})$ constitutes an optional use of the similarity score as a multiplicand:

$$\hat{p}(c_i) = (1 - \alpha) * \frac{1}{|A|} + \alpha * \sum_{j=1} \frac{\hat{p}(c_j)}{|P_j|} * sim(c_i, p_{jk}) \tag{6}$$

As the similarity scores were saved for all edges of the argument graph, different versions of the graph were constructed based on a minimum required similarity. Each version of the graph was then combined with different values of the weighting factor α and one of two versions of the recursive relevance (similarity weighted or not) to conduct a grid search across the combinations. While the two best combinations (minimum similarity $= 0.75$ [0.80], $\alpha = 0.3$ [0.4], both with unweighted recursive relevance) managed to achieve a slightly higher overall nDCG@10 than the blocklist-model (0.6944 [0.6924]), this does not permit the conclusion that ArgRank improved the retrieval performance of our model. Firstly, the reranking through ArgRank was applied on top of the blocklist-model and thus benefits from all previous approaches. Secondly, the improvement in nDCG resulted not from an increase in the relevance metric but from one in the "argument representation" metric by retrieving two new sentence pairs while the remaining 98 results stayed the same as for the blocklist model. Hence, the reranking did not improve the metric it aimed for and the observed improvement can only be attributed to chance.

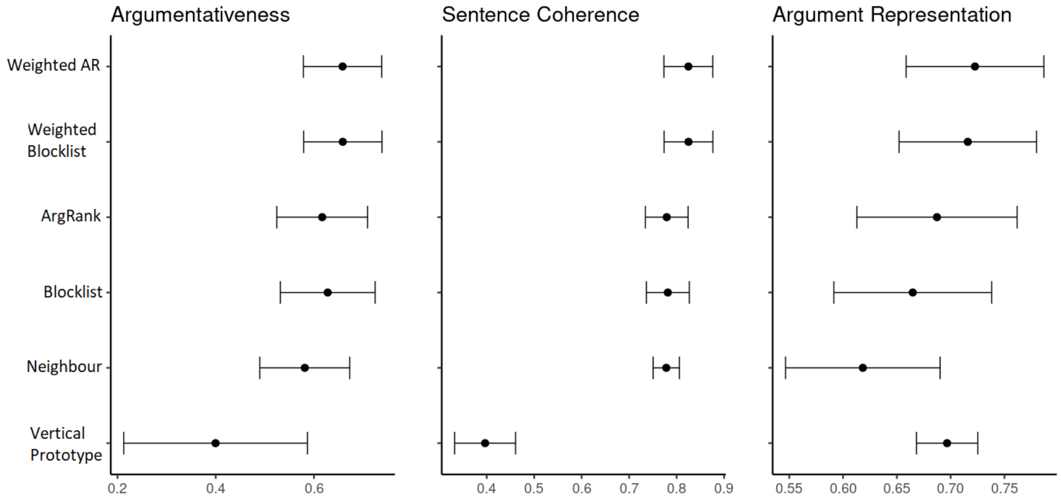

Figure 3: Mean nDCG@10 scores and variance per evaluation metric.

Model	Mean	Argumentativ.	Sentence Coherence	Argument Representation
Weighted AR	**0.7352** (0,00704)	0.6583 (0,07973)	0.8247 (0,05149)	**0.7226** (0,06399)
Weighted BL	0.7332 (0,00713)	**0.6587** (0,07978)	**0.8249** (0,05141)	0.7159 (0,06385)
ArgRank	0.6944 (0,00663)	0.6168 (0,09242)	0.7792 (0,04494)	0.6873 (0,07444)
Blocklist	0.6914 (0,00641)	0.6281 (0,09655)	0.7814 (0,04521)	0.6648 (0,07329)
Neighbour	0.6593 (0,01093)	0.5814 (0,09155)	0.7782 (0,02767)	0.6184 (0,07180)
Vertical P.	0.4977 (0,02971)	0.3997 (0,18709)	0.3966 (0,06420)	0.6967 (0,02848)

Table 1
Metric-specific and overall mean nDCG@10 with variance in parentheses.

4. Evaluation

The evaluation of our different approaches is based on the three following criteria with scores taken from $\{-2, 0, 1, 2, 3\}$: *Argumentativeness*, *Sentence Coherence* and *Argument Representation*. While the values between one and three were given for increasing quality in the respective metric, the meaning of both zero and -2 depended on the category. In order of the metrics above, a zero was used for (i) non-argumentative sentences, (ii) unrelated sentence pairs, and (iii) representation of the argument by only one sentence. The negative score was assigned to pairs that were (i) irrelevant to the query, (ii) contradicting themselves, and (iii) contradicting their parent argument. Before calculating the nDCG-scores, a value of two was added to all evaluations to shift the range to non-negative values. In all evaluation steps, the ten highest ranked results per query were evaluated by two persons at a time and the final score per sentence pair and metric calculated as the average of those two evaluations.
As discussed in chapter 3.2, the vertical prototype was evaluated on a larger set of 35 queries

to get a broader overview. The query set for all subsequent approaches was then restricted to the five best performing and worst performing queries of the vertical prototype. Hence, the values reported for the three metrics in Figure 3 are the average nDCG@10-scores across ten queries and two evaluators. The overall score is finally calculated as the average of the three nDCG-scores for each individual metric.

As visible in the graph, the introduction of a blocklist was able to improve performance beyond the neighbour-matching both in terms of argumentativeness as well as argument representation. Furthermore, the addition of ArgRank-based reranking only leads to slight improvements over the blocklist in the argument representation metric. Finally, using a weighted sum of the two relevance scores instead of ranking results by argument relevance first and sentence relevance second led to improvements both when using the ArgRank (Weighted AR vs. ArgRank) and when not using it (Weighted Blocklist vs. Blocklist).

5. Discussion

A few things became evident throughout the different stages of model development. Firstly, the DPH model's strength in identifying terms with a high information value for documents can turn into a weakness under certain circumstances. As shown in chapter 3.2, the model is susceptible to homonyms and does not regard the context a term is placed in. While we were able to partially address this issue with changes to the document collection and relevance scoring, a central weakness in the scope of this paper remains: Out of two documents with an equal number of occurrences f_{ij} for term j, DPH will reward the one with a lower number of terms $|d_i|$. Given that the relative importance of $|d_i|$ in equation 3 increases as documents have fewer terms, this effect becomes stronger for short documents such as sentences and potentially leads to biased results. Unfortunately, Dirichlet does not appear to be well-suited for sentence retrieval either, because the model's confidence in a document-specific language model decreases with document length. In order to clarify this assumption, a potential avenue for future research is to evaluate how well Dirichlet performs on the retrieval of argumentative sentences in comparison with DPH.

Secondly, we found naive solutions such as the introduction of a blocklist or "neighbour"-matching to have considerable positive impact on retrieval performance. While the success of our specific blocklist can be attributed to the origins of the document collection being debate platforms, we are confident that a similar solution is likely to also perform well on a broader corpus of argumentative sentences as recapping an opponents argument happens not exclusively on these platforms.

The finding that matching sentences with their immediate neighbours in parent arguments outperforms more sophisticated approaches can likely be explained by the capabilities of current language systems. While transformers like BERT prove very effective at encoding semantic information, they appear to not yet be on par with human debaters in the task of creating argumentatively sound pairs of sentences.

Finally, we found our version of the ArgRank to not lead to noticeable changes in retrieval performance despite slight improvements to the nDCG. This outcome can be attributed to the way the argument graph was constructed. Restricting the search for edges to unique conclusions and reuses in the same discussion with the same stance led to a sparse graph with only 44,250 edges for cosine similarity > 0.7. As a consequence, a maximum of only 10,806 of the total 302,388 arguments is rewarded by a higher ArgRank. Hence, to get a better understanding of the impact of ArgRank, we suggest that future research increases the search space for reuses of conclusions. A potential way to do this with the args.me-corpus is to find candidate arguments not only in the same discussion as argument i but in a set of retrieval results obtained by using the discussion title of i as a query. This approach, while likely to yield more "reuse candidates", introduces further uncertainty. On the one hand, potential reuses need to be more carefully evaluated to ensure that a candidate argument does in fact discuss the same topic as argument i. On the other hand, it needs to be determined if a candidate argument has the same stance as i or an opposing one, a task that continues to be difficult to solve in a domain-agnostic, automated way [29].

6. Conclusion

The retrieval model we proposed in our paper is a first step towards addressing the challenge of presenting short overviews of arguments on a controversial topic without omitting considerations like argumentative quality and logical coherence. The combination of two retrieval pipelines helps in retrieving sentence pairs that are not only deemed relevant by themselves but also originate from a set of relevant arguments. By applying two stages of quality filters, we further refine the retrieval results and remove arguments and sentences that are not of sufficient argumentative quality. Finally, by evaluating different matching approaches, we were able to increase logical coherence and argument representation of retrieved sentence pairs beyond the baseline of matching with an argument's conclusion.

Taken together, the stages of our retrieval model can offer key messages of arguments relevant to a user's information need. These sentence pairs are, however, by themselves insufficient for the opinion formation process and should only be used in combination with links to their sources to allow users to get a better understanding of the parent arguments.

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] H. Wachsmuth, B. Stein, Y. Ajjour, "PageRank" for Argument Relevance, in: P. Blunsom, A. Koller, M. Lapata (Eds.), 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017), Association for Computational Linguistics, 2017, pp. 1116–1126. URL: http://aclweb.org/anthology/E17-1105.

[3] 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: M. Hagen, S. Verberne, C. Macdonald, C. Seifert, K. Balog, K. Nørvåg,

V. Setty (Eds.), Advances in Information Retrieval. 44th European Conference on IR Research (ECIR 2022), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2022.

[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: K. Ashley, C. Cardie, N. Green, I. Gurevych, I. Habernal, D. Litman, G. Petasis, C. Reed, N. Slonim, V. Walker (Eds.), 4th Workshop on Argument Mining (ArgMining 2017) at EMNLP, Association for Computational Linguistics, 2017, pp. 49–59. URL: https://www.aclweb.org/anthology/W17-5106.

[5] I. Rahwan, F. Zablith, C. Reed, Laying the foundations for a world wide argument web, Artif. Intell. 171 (2007) 897–921.

[6] B. Stein, T. Gollub, M. Anderka, Retrieval Models, in: R. Alhajj, J. Rokne (Eds.), Encyclopedia of Social Network Analysis and Mining (ESNAM), Springer, Berlin Heidelberg New York, 2017, pp. 1–7. doi:10.1007/978-1-4614-7163-9_117-1.

[7] M. Potthast, L. Gienapp, F. Euchner, N. Heilenkötter, N. Weidmann, H. Wachsmuth, B. Stein, M. Hagen, Argument Search: Assessing Argument Relevance, in: 42nd International ACM Conference on Research and Development in Information Retrieval (SIGIR 2019), ACM, 2019. URL: http://doi.acm.org/10.1145/3331184.3331327. doi:10.1145/3331184.3331327.

[8] G. Amati, Frequentist and bayesian approach to information retrieval, in: ECIR, 2006.

[9] C. Zhai, J. Lafferty, A study of smoothing methods for language models applied to ad hoc information retrieval, in: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '01, Association for Computing Machinery, New York, NY, USA, 2001, p. 334–342. URL: https://doi.org/10.1145/383952.384019. doi:10.1145/383952.384019.

[10] S. E. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, M. Gatford, Okapi at trec-3, in: TREC, 1994.

[11] K. Spärck Jones, A statistical interpretation of term specificity and its application in retrieval, Journal of Documentation 28 (1972) 11–21. URL: https://doi.org/10.1108/eb026526.

[12] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web., Technical Report 1999-66, Stanford InfoLab, 1999. URL: http://ilpubs.stanford.edu:8090/422/, previous number = SIDL-WP-1999-0120.

[13] R. Johnson, J. Blair, Logical Self-defense, Key titles in rhetoric, argumentation, and debate series, International Debate Education Association, 2006. URL: https://books.google.de/books?id=ojNbr4vYooQC.

[14] F. van Eemeren, Reasonableness and Effectiveness in Argumentative Discourse: Fifty Contributions to the Development of Pragma-Dialectics, Argumentation Library, Springer International Publishing, 2015. URL: https://books.google.de/books?id=b1h1CgAAQBAJ.

[15] H. Wachsmuth, N. Naderi, Y. Hou, Y. Bilu, V. Prabhakaran, T. A. Thijm, G. Hirst, B. Stein, Computational Argumentation Quality Assessment in Natural Language, in: P. Blunsom, A. Koller, M. Lapata (Eds.), 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017), 2017, pp. 176–187. URL: http://aclweb.org/anthology/E17-1017.

[16] L. Gienapp, B. Stein, M. Hagen, M. Potthast, Efficient pairwise annotation of argument

quality, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 5772–5781. URL: https://aclanthology.org/2020.acl-main.511. doi:10.18653/v1/2020.acl-main.511.

[17] T. Green, L. Moroldo, A. Valente, Exploring bert synonyms and quality prediction for argument retrieval, in: [30], 2021, pp. 2374–2388. URL: http://ceur-ws.org/Vol-2936/#paper-213.

[18] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. arXiv:1810.04805.

[19] 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, CoRR abs/1911.11408 (2019). URL: http://arxiv.org/abs/1911.11408. arXiv:1911.11408.

[20] Y. Ajjour, H. Wachsmuth, J. Kiesel, M. Potthast, M. Hagen, B. Stein, Data Acquisition for Argument Search: The args.me corpus, in: C. Benzmüller, H. Stuckenschmidt (Eds.), 42nd German Conference on Artificial Intelligence (KI 2019), Springer, Berlin Heidelberg New York, 2019, pp. 48–59. doi:10.1007/978-3-030-30179-8_4.

[21] L. Gienapp, Quality-aware argument retrieval with topical clustering, in: [30], 2021, pp. 2366–2373. URL: http://ceur-ws.org/Vol-2936/#paper-212.

[22] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, C. Lioma, Terrier: A high performance and scalable information retrieval platform (2006).

[23] 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.

[24] G. Amati, Probability models for information retrieval based on divergence from randomness, 2003.

[25] D. Pal, M. Mitra, K. Datta, Query expansion using term distribution and term association, 2013. arXiv:1303.0667.

[26] A. Shukla, S. K. Das, Bose einstein 1 and bose einstein 2 model for optimal query expansion, 2020.

[27] J. Carbonell, J. Goldstein, The use of mmr, diversity-based reranking for reordering documents and producing summaries, 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. 335–336. URL: https://doi.org/10.1145/290941.291025. doi:10.1145/290941.291025.

[28] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, Mpnet: Masked and permuted pre-training for language understanding, 2020. arXiv:2004.09297.

[29] E. Körner, G. Wiedemann, A. D. Hakimi, G. Heyer, M. Potthast, On Classifying whether Two Texts are on the Same Side of an Argument, in: The 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021), Association for Computational Linguistics, 2021, pp. 10130–10138. URL: https://aclanthology.org/2021.emnlp-main.795/.

[30] G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum (CLEF 2021), number 2936 in CEUR Workshop Proceedings, Aachen, 2021. URL: http://ceur-ws.org/Vol-2936/.