

Grimjack at Touché 2022: Axiomatic Re-ranking and Query Reformulation

Notebook for the Touché Lab on Argument Retrieval at CLEF 2022

Jan Heinrich Reimer, Johannes Huck and Alexander Bondarenko

Martin-Luther-Universität Halle-Wittenberg, 06099 Halle (Saale), Germany

Abstract

In this paper, we present the Team's Grimjack retrieval approaches for the Touché shared task on Argument Retrieval for Comparative Questions. In total, we submit five runs that pursue the two main objectives: favoring argumentative and high argument quality documents in the final ranking and balancing stance-based exposure by ensuring an even ratio of pro and con arguments at top ranks. Our results indicate that BM25 outperforms query likelihood ranking for initial passage retrieval and that stance-based re-ranking can slightly improve a ranking effectiveness. For stance classification, prompting the T0 zero-shot language model is the best-performing approach when considering all available ground-truth labels.

Keywords

Axiomatic Re-ranking, Query Reformulation, Comparative Questions, Argument Quality, Argument Stance

1. Introduction

Argument retrieval is a specific task that not only considers topical relevance of retrieved documents to given queries (usually of controversial, argumentative, or opinion nature) but also accounts for argument specific features like argument quality and stance [1, 2]. Furthermore, it has been shown that current search engines might return biased results [3] and argument retrieval systems return imbalanced pro / con arguments [4]. We especially emphasize the importance of retrieving diverse results for comparative questions (e.g., "Train or plane? Which is the better choice?") that provide different point of views to mitigate biasing users' decisions towards one or the other comparison option.


Our Team Grimjack participated in the Touché shared task on Argument Retrieval for Comparative Questions which goals are: (1) To retrieve relevant and high quality argumentative passages from a collection of 868 655 text passages to a set of 50 search topics and (2) to classify the stance of the retrieved passages towards the comparison objects in search topics [5]. As part of our participation in the task, we have developed a flexible retrieval pipeline in Python based


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

✉ jan.reimer@student.uni-halle.de (J. H. Reimer); johannes.huck@student.uni-halle.de (J. Huck);
alexander.bondarenko@informatik.uni-halle.de (A. Bondarenko)

🌐 <https://heinrich.reimer.family> (J. H. Reimer); <https://github.com/johannes-huck> (J. Huck);
<https://sites.google.com/view/alexanderbondarenko> (A. Bondarenko)

🆔 0000-0003-1992-8696 (J. H. Reimer); 0000-0002-1678-0094 (A. Bondarenko)

 © 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

on Pyserini [6] as an easily configurable command line application, which we release under a free open source license.¹ In the first step, our approach uses query (comparative questions from topics’ titles) reformulation and expansion by important terms from topics’ descriptions and narratives. Then the top-10 initially retrieved passages using query likelihood with Dirichlet smoothing [7] are axiomatically re-ranked based on the number and position of premises, claims (identified with TARGER [8]), and comparison objects, and argument quality predictions by the IBM Debater API [9] and T0++ [10]. Finally, the pro and con argumentative passages towards the compared objects are balanced in the final ranking by alternating documents of different stance (cf. Section 3 for more details on the approach and submitted runs). We also submitted our software using the TIRA platform [11]² that automatically evaluates submitted approaches and presents the results on a leaderboard.

Even though none of our runs (with query likelihood first-stage retrieval) outperform the official BM25 baseline in terms of relevance and rhetorical quality, we observe that stance-based re-ranking can slightly improve a ranking effectiveness while argument axiomatic re-ranking with KWIKSORT does not change retrieval effectiveness. Our runs using query expansion with the T0++ language model [10] should pose examples to discuss current doubts about the usefulness of large zero-shot language models in the field of search and information retrieval [3] as they are amongst the worst performing runs. For stance classification however, our T0-based approach using zero-shot prompts yields promising results, even though we are unable to directly compare it to other runs due to different test set coverage.

2. Related Work

Personal decision making often starts with formulating comparative questions like “Should I major in philosophy or psychology?” [1, 2, 5]. Short direct answers (potentially biased) [12] to such questions might be insufficient; instead, such questions require diverse opinions to provide a sufficient, balanced, and argumentative overview [1]. The Touché shared task on Argument Retrieval for Comparative Questions was proposed to evaluate retrieval approaches on a large corpus with respect to relevance and rhetorical quality of potential answers to comparative questions that also may represent different standpoints [5, 13].

The most effective approaches at previous Touché editions [1, 2] successfully used query expansion with synonyms and antonyms [14], identified premises and claims in retrieved documents [15, 16], estimated argument quality [14], and re-ranked initially retrieved documents based on argument quality and document “comparativeness”, e.g., a ratio of comparative adjectives [17]. Inspired by the participant approaches from the previous Touché editions, we also include the components of argument mining and argument quality estimation in our retrieval pipeline, however, using different methods. We rely on a large language model T0 trained in multitask setting that showed to achieve state-of-the-art results for various Natural Language Processing tasks in zero-shot settings [10]. The largest pretrained T0 variant, T0++, was trained on 62 datasets with 12 task-specific prompts covering such tasks as question answering, sentiment analysis, summarization, etc. By using T0++, we aim for answering a question whether

¹<https://github.com/heinrichreimer/grimjack>

²<https://tira.io>

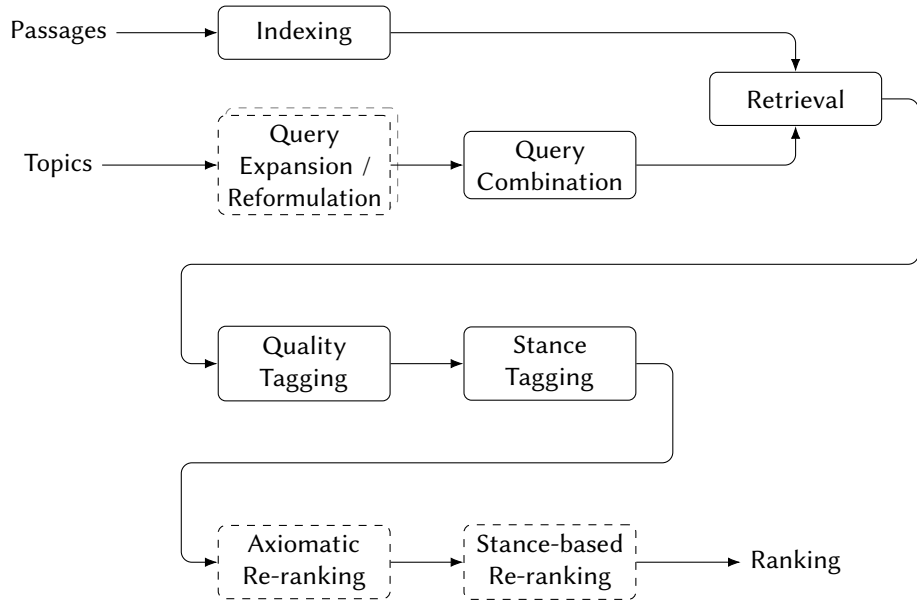


Figure 1: Architecture overview for the modular retrieval pipeline used to produce our runs. Dashed boxes indicate optional steps, that are not used in all runs.

the abilities of large language models are sufficient for the new task of argument retrieval.

Our second idea of axiomatic re-ranking comes from axiomatic thinking in information retrieval, where axioms formally describe constraints that good retrieval model should fulfil, e.g., documents with more query term occurrences should be ranked higher Fang et al. [18]. It has already been shown that combining multiple axioms for re-ranking results of arbitrary retrieval models can improve final overall retrieval effectiveness [19]. Complementing existing retrieval axioms, Bondarenko et al. [20] introduced argumentativeness axioms based on claims and premises in documents identified with TARGER [8].

3. Approach

We design the architecture of our argumentative retrieval system as a multi-step pipeline that subsequently (re-)ranks, annotates, or modifies documents retrieved for each query with the query likelihood with Dirichlet smoothing ($\mu = 1\ 000$). As shown in Figure 1, our proposed pipeline consists of four main steps: (1) query expansion, reformulation, and combination, (2) first-stage retrieval, (3) argument quality estimation and stance detection, and (4) axiomatic re-ranking and stance-based re-ranking.

3.1. Query Expansion, Reformulation, and Combination

The first step of our retrieval pipeline is original query (task’s topic titles) reformulation and expansion that aims for increasing a recall. For that, we use two different strategies: (1) replacing the comparison objects with their synonyms (e.g., Ubuntu vs. Windows \rightarrow Linux vs. Windows)

Table 1

Original queries (topic titles) provided by Touché and generated queries by T0++ [10] by prompting the topic’s description (D) or narrative (N).

Topic	Original query	Field	Generated query
12	Train or plane? Which is the better choice?	D	Travel
		N	What are the benefits of trains over planes for inter-continental travel?
53	Should I buy steel or ceramic knives?	D	Why should I choose ceramic knives over steel knives?
		N	What are the pros and cons of ceramic knives?
88	Should I major in philosophy or psychology?	D	What is the difference between philosophy and psychology?
		N	What are the benefits of a major in English or history?
95	Which is more environmentally friendly, a hybrid or a diesel?	D	What are the most environmentally friendly cars?
		N	What is more environmentally friendly, a diesel or a hybrid car?

and (2) generating additional, new queries exploiting the topics’ description and narrative provided by the task organizers [5]. We then address the precision-recall trade-off by deploying re-ranking steps by moving more relevant documents at the top of the ranking (cf. Section 3.4).

Query Reformulation with Synonyms. To find synonyms of comparison objects mentioned in questions (search queries), we use two different strategies: (1) word embeddings and (2) a zero-shot generation with pre-trained large language models. For the first strategy, we use fastText word embeddings [21] from PyMagnitude³ to find the word with the highest cosine similarity to the given comparison objects in the embedding space. We manually examine synonyms from the fastText embeddings pre-trained on different corpora (e.g., Wikipedia and Twitter) and find that the Twitter-based embeddings provide more accurate synonyms.

Our second strategy is based on the T0++ zero-shot language model [10]. We prompt the model to generate an answer to the following question: What are synonyms of the word <token>?, where <token> is one of the two original comparison objects. We then process the output by splitting by commas and select the first term that is different from the original query term. With the synonyms returned by either strategy, we replace the comparison objects to formulate new question queries.

Query Reformulation with Topic Context. In the next step, we complement the expanded queries with two newly generated ones per topic taking into account the contextual information from the topic’s descriptions and narratives that contain important details on the actual information. Using the Hugging Face Inference API [22], we prompt T0++

³<https://gitlab.com/Plasticity/magnitude>

with the following task: <text>. Extract a natural search query from this description., where <text> is either the topic’s narrative or description. In Table 1, we show the examples of generated queries. Albeit some of the generated queries (e.g., topic 53) are just reformulations of the original one, T0++ also generates potentially useful meaningful new queries (e.g., topic 12).

Query Combination and Expansion. Finally, we combine up to 5 question queries (reformulated with synonyms, generated, and the original one, depending on the submitted run; cf. Section 4) using a logical disjunction (Pyserini’s OR operator). We choose the logical disjunction with the outlook on increasing the system’s recall and decreasing the chance of empty result sets in the case that search terms are not present in the corpus.

In total we submitted 5 runs (retrieval results; cf. Section 4) to the task, in some of which we use only the original query, and the expanded queries in the others to test the influence of the query expansion and reformulation on the final ranking results.

3.2. Passage Retrieval

To retrieve passages from the task’s corpus, we first build an inverted index using the Pyserini framework [6]. In the index, we store index term positions, passage vectors, and raw passage contents. Index terms are stemmed using the Porter stemmer [23] and stop words are removed as per the default Pyserini stopword list [6]. We then retrieve passages for the previously combined query (cf. Section 3.1) using the query likelihood model with Dirichlet smoothing ($\mu = 1\,000$). From this first-stage ranker, we retrieve 100 candidate passages for each query.

3.3. Argument Tagging, Argument Quality and Stance Classification

After retrieving candidate passages, we tag the argumentative structure (premises and claims), estimate argument quality, and detect the stance (whether the passage is pro first comparison object, pro second, has neutral, or no stance.). This information is used in later steps of our retrieval pipeline for re-ranking (cf. Section 3.4). We tag each passage’s argumentative structure with the TARGER argument tagger [8] using the `targer-api` Python package⁴.

To estimate the passage’s argument quality and detect the stance, we first split each passage into sentences using the NLTK library [24]. Then each sentence is treated as one potential argument; the quality score and stance for the whole passage is calculated by averaging the quality or stance scores for all sentences in the passage.

Argument Quality Estimation. We use two different methods for assessing the argument quality. Our first method is based on the IBM Debater API [9].⁵ The API then determines how good the quality of each argument with regard to the topic is with a BERT-based [25] regression classifier model trained on the IBM-ArgQ-6.3kArgs dataset. The API returns a quality score ranging from 0 (low quality) to 1 (high quality).

⁴<https://github.com/webis-de/targer-api>

⁵<https://early-access-program.debater.res.ibm.com>

Table 2

Argument quality label and argument stance label mapping from textual tokens returned T0++ [10].

- (a) Argument quality label mapping for the prompt
How would you rate the readability and consistency in this sentence?.
- (b) Argument stance label mapping for the prompts
Is this sentence pro <object>? (Pro) and
Is this sentence against <object>? (Con)
given a single comparative object <object>.

Text Label	Value
very good	1.00
good	0.75
bad	0.25
very bad	0.00
other	0.50

Text Label		Value
Pro	Con	
yes / pro	yes / con	0
yes / pro	no	+1
no	yes / con	-1
no	no	0
other	other	0

As a second method to obtain the argument quality we also use the T0++ model [10] and prompt it to generate a text to the following task: <sentence>. How would you rate the readability and consistency in this sentence? very good, good, bad, very bad, where <sentence> is one of the passage sentences. We then map the models textual outputs to numeric values using the mapping shown in Table 2a.

Stance Detection. Stance detection for each sentence uses the same conceptual approaches but with different inputs and outputs. Since both the IBM Debater API [26] and T0++ [10] can predict only a single-target stance (i.e., for one of the two comparison objects), we combine the two single-target stance scores into a multi-target stance by taking the difference between the stance towards the first object and the stance towards the second object. We also experimented with different thresholds for the minimal difference between the single-target stances and found a threshold of 0.125 to work well by manually examining some classified examples.

For scoring the single-target argument stance for a sentence with the IBM Debater API, we again query the API with a sentence (argument) and a topic created using one of the comparison objects. The classifier [26] then computes an argument’s likelihood of being pro, con, or neutral with respect to the topic (i.e., the comparison object in our pipeline) by first classifying a sentiment and then detecting whether the topic’s and argument’s targets contradict each other. The API then returns a score from from -1 (against the comparison object) to +1 (in favor). By classifying different topics for each object (i.e., <object> is good and <object> is the best), we determine an averaged single-target stance for each comparison object.

When using the T0++ for the stance detection, we first experiment with directly prompting the model to output ‘pro’, ‘con’, or ‘neutral’ labels for the comparison objects. We formulate the task as two simple questions passed to the model, one to determine whether the sentence has a positive stance towards the comparison object and one to determine whether it has a negative stance: <sentence> Is this sentence pro <object>? yes or no and <sentence> Is this sentence against <object>? yes or no, where <sentence> is one sentence of the passage and <object> is one of the comparative objects. This results in two answers (yes

Table 3

Axioms used in our retrieval pipeline. An asterisk (*) indicates newly proposed axioms.

Name	Description
ArgUC [20]	Prefer more argumentative units.
QTArg [20]	Prefer more query terms in argumentative units.
QTPArg [20]	Prefer earlier query terms in argumentative units.
CompArg*	Prefer more comparative objects in argumentative units.
CompPArg*	Prefer earlier comparative objects in argumentative units.
aSLDoc [27]	Prefer passages with 12–20 words per sentence.
ArgQ*	Prefer higher argument quality.

or no) for the positive and negative stance respectively. We combine the two textual answers using the mapping shown in Table 2b.

3.4. Axiomatic and Stance-based Re-rankers

Since recall of our retrieval system is increased by expanding and reformulating queries (cf. Section 3.1), we seek to improve precision by re-ranking the top-10 passages from the first-stage retrieval (cf. Section 3.2) using two different strategies that should rank more argumentative and of higher quality passages also ensuring a balanced overview of the two comparison objects. (1) We re-rank based on argumentativeness axioms, and (2) we re-rank based on the passages’ stances towards the comparison objects.

Argumentative Axiomatic Re-ranking. Ranking methods such as BM25 or query likelihood with Dirichlet smoothing do not capture the “argumentativeness” in text that is important for argument retrieval [5]. Some approaches for at the TREC Common Core and Decision tracks exploit task-specific, argumentativeness axioms to address the document argumentativeness [20, 27]. Axioms are constraints that define pairwise ranking preferences between documents or passages. Because of the promising development in the field of axiomatic information retrieval [28], we re-rank the top-10 initially retrieved passages with the KWIKSORT algorithm [19]. For axiomatic re-ranking, we compute preferences for 7 argumentativeness axioms specified in Table 3. The axioms cover general argumentativeness (ArgUC), argumentative relevance (QTArg, QTPArg), comparative relevance (CompArg, CompPArg), and rhetorical and argumentative quality (aSLDoc, ArgQ). We then combine the axioms in a majority voting scheme, i.e., we only keep preferences where at least 50 % of the 7 axioms agree, and fall back to the original ranking order if less than 50 % of all axioms agree. Using the `ir_axioms` framework [28],⁶ we then re-rank with the combined axiom.

Stance-based Re-ranking. We also implement a stance-based re-ranker to produce rankings where the two conflicting stances (pro first comparison object and pro second comparison object) are nearly equally present. For balancing the stances, we experiment with two different

⁶https://github.com/webis-de/ir_axioms/

re-ranking strategies: (1) alternating stance and (2) balanced top- k stance. For the alternating stance strategy, we split the result set into three lists: (1) with arguments in favor of the first comparison object, (2) in favor of the second comparison object, and (3) neutral arguments or arguments with no stance. We then alternately select passages from the first two lists. If one or both lists are empty, we fall back to the neutral list. The balanced top- k stance strategy is based on the original ranking. Here we count the number of passages in favor of the first comparison object and the second comparison object in the top- k initially retrieved passages. If the difference of these two values is greater than 1, we move the last passage from the majority within the top- k ranking behind the first minority passage after the top- k ranking. This way, passages of the underrepresented stance advance the ranking until the ranking is balanced in the top- k positions. In initial experiments, however, we find the alternating stance strategy to be more promising, because the balanced top- k stance strategy often lead to rankings containing mostly neutral passages.

4. Submitted Runs

We submit five runs that use different components and strategies of our pipeline (cf. Section 3) to the Touché second task. Instead of uploading generated run files, we deploy our retrieval system as a working software on the TIRA platform [11].

Query Likelihood Baseline (Run 1). For our first run, we simply retrieve top-100 passages ranked by query likelihood with Dirichlet smoothing [7] ($\mu = 1000$) for the original, unmodified queries (topic titles) and tag argument stance by comparing sentiments for each object using the IBM Debater API, treating a stance under a threshold of 0.125 as neutral.

Argument Axioms (Run 2). To produce our second run, we re-rank the top-10 passages from the baseline result using KWIKSORT [28, 19] based on preferences from the argument axioms as described in Section 3.4.

Stance-based Re-ranking with Argumentative Axioms (Run 3). Our third run also uses argument axiomatic re-ranking after the baseline retrieval. However to ensure that the stances towards both comparison objects are nearly equally represented in the result ranking, we apply stance-based re-ranking with the alternating stance strategy as described in Section 3.4.

All You Need is T0 (Run 4). Large language models have recently found application in many NLP tasks, web search, or retrieval. The trend of using large language models for solving almost any task has also been criticized. For instance, Shah and Bender [3] highlight conceptual flaws that question if such an extreme usage of not fully understood models is desirable when implementing search for answers to real-life questions (e.g., in search engines).

In our fourth submitted run, we want test a language model’s T0++ zero-shot classification abilities. First, we reformulate and generate and combine queries; final queries are an expansion of the topic titles (cf. Section 3.1). We then retrieve 100 documents using query likelihood, and use T0++ again to estimate argument quality and stance (cf. Section 3.3).

Table 4

Relevance results of selected runs submitted to Task 2: Argument Retrieval for Comparative Questions. Reported are the mean nDCG@5 and the 95% confidence intervals for our runs, the best task’s run result (team Captain Levi), and the official task baseline (Puss in Boots, in italics).

Team	Run	nDCG@5		
		Mean	Low	High
Captain Levi [29]	dense_initial_retr.	0.758	0.708	0.810
<i>Puss in Boots</i> [5]	<i>BM25-Baseline</i>	<i>0.469</i>	<i>0.403</i>	<i>0.535</i>
Grimjack	Run 3	0.422	0.349	0.500
Grimjack	Run 2	0.376	0.299	0.455
Grimjack	Run 1	0.376	0.301	0.459
Grimjack	Run 5	0.349	0.270	0.425
Grimjack	Run 4	0.345	0.273	0.425

Argumentative Stance-based Re-ranking with T0 (Run 5). In our last run, we combine most of the methods introduced in Section 3 to generate a ranking that is both as argumentative as possible and equally represents argument stances, but also uses T0++ for query reformulation and expansion. Here, we combine new queries generated by T0++ and reformulate queries by replacing synonyms returned by T0++. However, we also use synonyms from the fastText [21] embedding similarity method (cf. Section 3.1); final queries are an expansion of the topic titles. The top-10 results of the 100 passages retrieved using query likelihood for the expanded queries are then re-ranked based on the argumentativeness axioms and by alternating stance (cf. Section 3.4).

5. Results

We evaluate our approach by effectiveness to retrieve relevant and high-quality passages and to predict the correct stance towards the comparison objects, using manual judgments provided by Touché. The task organizers asked human volunteers to label each document pooled from all submitted runs at depth 5 with respect to relevance (0: not relevant, 1: relevant, 2: highly relevant), rhetorical quality (0: low quality or not argumentative, 1: average quality, 2: high quality), and stance (pro first object, pro second object, neutral, no stance).

The results for the relevance and quality effectiveness using nDCG@5 (Tables 4 and 5) show that our baseline Run 1 using query likelihood with Dirichlet smoothing performs worse than the BM25 baseline (Puss in Boots [5]). Since our other runs re-rank retrieved results from the initial ranking, we compare our individual re-ranking strategies. Nonetheless, we acknowledge that all of our submitted runs are outperformed by the BM25 baseline and other dense rankers’ results submitted to the shared task. The differences in nDCG@5 scores compared to our query likelihood baseline indicate that axiomatic re-ranking (Run 2) can increase consistency with argumentativeness axioms while retaining equal retrieval effectiveness. Unfortunately, query expansion with T0++ slightly decreases nDCG@5 on average by about 3 p.p. for relevance judgments and 2 p.p. for quality judgments. Stance-based re-ranking, however, can increase

Table 5

Quality results of selected runs submitted to Task 2: Argument Retrieval for Comparative Questions. Reported are the mean nDCG@5 and the 95% confidence intervals for our runs, the best task’s run result (team Aldo Nadi), and the official task baseline (Puss in Boots, in italics).

Team	Run	nDCG@5		
		Mean	Low	High
Aldo Nadi [30]	RF_reranked	0.774	0.719	0.828
<i>Puss in Boots</i> [5]	<i>BM25-Baseline</i>	<i>0.476</i>	<i>0.400</i>	<i>0.553</i>
Grimjack	Run 3	0.403	0.331	0.478
Grimjack	Run 5	0.365	0.290	0.445
Grimjack	Run 2	0.363	0.289	0.442
Grimjack	Run 1	0.363	0.287	0.443
Grimjack	Run 4	0.344	0.266	0.428

Table 6

Stance detection results of selected runs submitted to Task 2: Argument Retrieval for Comparative Questions. Reported are a macro-averaged F_1 score and number of documents N where the predicted stance has a ground-truth label for our runs, the best task’s run result (team Captain Levi), and the official task baseline that always predicts ‘no stance’ (Puss in Boots, in italics). F_1 score is computed for all predicted stance labels with corresponding ground-truth labels (All) or only for the top-5 passages per run (Top-5).

Team	Run	All		Top-5	
		F_1	N	F_1	N
Grimjack	Run 4	0.313	1208	0.235	250
Captain Levi [29]	dense_initial_retr.	0.301	1688	0.359	250
Grimjack	Run 2	0.207	1282	0.180	250
Grimjack	Run 1	0.207	1282	0.180	250
Grimjack	Run 3	0.207	1282	0.175	250
Grimjack	Run 5	0.199	1180	0.168	250
<i>Puss In Boots</i> [5]	<i>Always-NO-Baseline</i>	<i>0.158</i>	<i>1328</i>	<i>0.159</i>	<i>250</i>

nDCG@5 by up to 5 p.p. for relevance judgments and by 4 p.p. for quality judgments. None of our re-ranking stages could sufficiently compensate for the worse retrieval performance of the initial query likelihood ranking.

For stance detection, we compare the T0-based stance classification approach with the best competing team’s approach (Captain Levi, pre-trained RoBERTa without fine-tuning) and the baseline (Puss in Boots) that predicts the majority class (‘no stance’). In Table 6, we report a macro-averaged F_1 -score per run and per team as well as the number of documents N for which the predicted stance has a ground-truth label as provided by the task organizers. We observe that since only the top-5 passages were pooled for manual judgments only a limited number of predicted stance labels (e.g. 1208 for Run 4) can be used for evaluation, even though we predicted the stance up to depth 100 (i.e. 5000 predicted stance labels per run). In this setting our Run 4 (i.e. stance prediction using T0++; cf. Section 3.3) has the highest macro-averaged F_1 -score of all

submitted runs to the task. However, due to the limited number of labels available for evaluation and because the number of available labels differs across teams and runs, we cannot directly compare different runs. For example, the 3 792 unjudged labels from Run 4 could be correctly predicted (i.e., increasing F_1) or incorrectly predicted (i.e., decreasing F_1). As an alternative, comparable measure, in the rightmost columns of Table 6, we report F_1 -scores of predicted stances of only the top-5 passages of each run. All 250 stance labels from the top-5 results of each submitted run have corresponding ground-truth labels due to the organizers’ top-5 pooling for manual judgment. When considering only the top-5 passages, our stance classification approach using T0++ falls behind Team Captain Levi’s best performing approaches. However, 250 samples might also be an insufficient sample size to compare classifier performance. It is also unclear how examining only top results affects the evaluation of classification performance.

6. Conclusion

In our approaches to retrieve relevant and high-quality argumentative passages that help answer comparative questions, we combine query reformulation and expansion techniques with axiomatic re-ranking exploiting argumentative structure and argument quality and stance. Using the IBM Debater API and the T0++ language model, we showcase two state-of-the-art approaches for argument quality estimation. We extend previous query expansion approaches used in the Touché shared tasks by incorporating the contextual information provided in topics’ descriptions and narratives. To attain nearly equal exposure across argument stances in the final ranking, we balance the pro and con arguments on top-10 ranks.

While none of our runs outperform the BM25 baseline in terms of nDCG@5 on relevance and quality judgments, we find that axiomatic re-ranking and stance-based re-ranking can slightly increase the effectiveness of the first-stage query likelihood ranking. This poses an interesting direction for future work: applying our proposed re-ranking strategies to results of other retrieval models, e.g., BM25. Since our run featuring query expansion with generated texts by T0++ is the worst-performing in terms of relevance and rhetorical quality, we also question the usefulness of large language models in early retrieval stages. Our results represent additional motivation to investigate the effect of explainability on retrieval performance, as recently questioned in the community.

Our approach to stance classification heuristically maps single-target stance classification results to multi-target, and we were not able to find a satisfactory strategy to distinguish neutral stance from passages without stance. Arguably, fine-tuning a multi-class neural classifier like BERT on the stance dataset provided by Touché could possibly improve classification performance by directly predicting the multi-target stance. Our evaluation of F_1 stance prediction performance yields no clear winner as the participating teams predicted stance labels for different, potentially biased sub-sets of the document collection resulting in different test set coverage. We encourage future work to reproduce and evaluate stance prediction approaches of all participating teams on an independent test dataset.

Acknowledgments

This work was partially supported by the Deutsche Forschungsgemeinschaft (DFG) through the project “ACQuA 2.0” (Answering Comparative Questions with Arguments; project number 376430233).

References

- [1] 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. Névéal (Eds.), Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020, volume 2696 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/paper_261.pdf.
- [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.), Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021, volume 2936 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 2258–2284. URL: <http://ceur-ws.org/Vol-2936/paper-205.pdf>.
- [3] C. Shah, E. M. Bender, Situating Search, in: D. Elswiler (Ed.), CHIIR '22: ACM SIGIR Conference on Human Information Interaction and Retrieval, Regensburg, Germany, March 14 - 18, 2022, ACM, 2022, pp. 221–232. URL: <https://doi.org/10.1145/3498366.3505816>.
- [4] S. P. Cherumanal, D. Spina, F. Scholer, W. B. Croft, Evaluating Fairness in Argument Retrieval, in: G. Demartini, G. Zuccon, J. S. Culpepper, Z. Huang, H. Tong (Eds.), CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021, ACM, 2021, pp. 3363–3367. URL: <https://doi.org/10.1145/3459637.3482099>.
- [5] 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.
- [6] J. Lin, X. Ma, S. Lin, J. Yang, R. Pradeep, R. Nogueira, Pyserini: An Easy-to-Use Python Toolkit to Support Replicable IR Research with Sparse and Dense Representations, CoRR abs/2102.10073 (2021). URL: <https://arxiv.org/abs/2102.10073>. arXiv:2102.10073.
- [7] C. Zhai, J. D. Lafferty, A Study of Smoothing Methods for Language Models Applied to Ad Hoc Information Retrieval, in: W. B. Croft, D. J. Harper, D. H. Kraft, J. Zobel (Eds.), SIGIR 2001: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, September 9-13, 2001, New Orleans, Louisiana, USA, ACM, 2001, pp. 334–342. URL: <https://doi.org/10.1145/383952.384019>.
- [8] A. N. Chernodub, O. Oliynyk, P. Heidenreich, A. Bondarenko, M. Hagen, C. Biemann, A. Panchenko, TARGER: Neural Argument Mining at Your Fingertips, in: M. R. Costa-

- jussà, E. Alfonseca (Eds.), Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations, Association for Computational Linguistics, 2019, pp. 195–200. URL: <https://doi.org/10.18653/v1/p19-3031>.
- [9] A. Toledo, S. Gretz, E. Cohen-Karlik, R. Friedman, E. Venezian, D. Lahav, M. Jacovi, R. Aharonov, N. Slonim, Automatic Argument Quality Assessment - New Datasets and Methods, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, Association for Computational Linguistics, 2019, pp. 5624–5634. URL: <https://doi.org/10.18653/v1/D19-1564>.
- [10] V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, T. L. Scao, A. Raja, M. Dey, M. S. Bari, C. Xu, U. Thakker, S. Sharma, E. Szczechla, T. Kim, G. Chhablani, N. V. Nayak, D. Datta, J. Chang, M. T. Jiang, H. Wang, M. Manica, S. Shen, Z. X. Yong, H. Pandey, R. Bawden, T. Wang, T. Neeraj, J. Rozen, A. Sharma, A. Santilli, T. Févry, J. A. Fries, R. Teehan, S. Biderman, L. Gao, T. Bers, T. Wolf, A. M. Rush, Multitask Prompted Training Enables Zero-Shot Task Generalization, CoRR abs/2110.08207 (2021). URL: <https://arxiv.org/abs/2110.08207>. arXiv:2110.08207.
- [11] 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 - Lessons Learned from 20 Years of CLEF, volume 41 of *The Information Retrieval Series*, Springer, 2019, pp. 123–160. URL: https://doi.org/10.1007/978-3-030-22948-1_5.
- [12] M. Potthast, M. Hagen, B. Stein, The Dilemma of the Direct Answer, SIGIR Forum 54 (2020) 14:1–14:12. URL: <https://doi.org/10.1145/3451964.3451978>.
- [13] A. Bondarenko, Y. Ajjour, V. Dittmar, N. Homann, P. Braslavski, M. Hagen, Towards Understanding and Answering Comparative Questions, in: K. S. Candan, H. Liu, L. Akoglu, X. L. Dong, J. Tang (Eds.), WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022, ACM, 2022, pp. 66–74. URL: <https://doi.org/10.1145/3488560.3498534>.
- [14] T. Abye, T. Sager, A. J. Triebel, An Open-Domain Web Search Engine for Answering Comparative Questions, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020, volume 2696 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/paper_130.pdf.
- [15] J. Huck, Development of a Search Engine to Answer Comparative Queries, in: L. Cappellato, C. Eickhoff, N. Ferro, A. Névéol (Eds.), Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020, volume 2696 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020. URL: http://ceur-ws.org/Vol-2696/paper_178.pdf.
- [16] E. Shirshakova, A. Wattar, Thor at Touché 2021: Argument Retrieval for Comparative Questions, in: 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, Bucharest, Romania, September 21st - to - 24th, 2021, volume 2936 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 2455–2462. URL: <http://ceur-ws.org/Vol-2936/paper-219.pdf>.

- [17] V. Chekalina, A. Panchenko, Retrieving Comparative Arguments using Ensemble Methods and BERT, in: 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, Bucharest, Romania, September 21st - to - 24th, 2021, volume 2936 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 2354–2365. URL: <http://ceur-ws.org/Vol-2936/paper-211.pdf>.
- [18] H. Fang, T. Tao, C. Zhai, A Formal Study of Information Retrieval Heuristics, in: M. Sanderson, K. Järvelin, J. Allan, P. Bruza (Eds.), SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004, ACM, 2004, pp. 49–56. URL: <https://doi.org/10.1145/1008992.1009004>.
- [19] M. Hagen, M. Völske, S. Göring, B. Stein, Axiomatic Result Re-Ranking, in: S. Mukhopadhyay, C. Zhai, E. Bertino, F. Crestani, J. Mostafa, J. Tang, L. Si, X. Zhou, Y. Chang, Y. Li, P. Sondhi (Eds.), Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016, ACM, 2016, pp. 721–730. URL: <https://doi.org/10.1145/2983323.2983704>.
- [20] A. Bondarenko, M. Hagen, M. Völske, B. Stein, A. Panchenko, C. Biemann, Webis at TREC 2018: Common core track, in: E. M. Voorhees, A. Ellis (Eds.), Proceedings of the Twenty-Seventh Text REtrieval Conference, TREC 2018, Gaithersburg, Maryland, USA, November 14-16, 2018, volume 500-331 of *NIST Special Publication*, National Institute of Standards and Technology (NIST), 2018. URL: <https://trec.nist.gov/pubs/trec27/papers/Webis-CC.pdf>.
- [21] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching Word Vectors with Subword Information, *Trans. Assoc. Comput. Linguistics* 5 (2017) 135–146. URL: <https://transacl.org/ojs/index.php/tacl/article/view/999>.
- [22] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-Art Natural Language Processing, in: Q. Liu, D. Schlangen (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 38–45. URL: <https://doi.org/10.18653/v1/2020.emnlp-demos.6>.
- [23] M. F. Porter, An Algorithm for Suffix Stripping, *Program* 14 (1980) 130–137. URL: <https://doi.org/10.1108/eb046814>.
- [24] S. Bird, E. Klein, E. Loper, *Natural Language Processing with Python*, O’Reilly, 2009. URL: <http://www.oreilly.de/catalog/9780596516499/index.html>.
- [25] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186. URL: <https://doi.org/10.18653/v1/n19-1423>.
- [26] R. Bar-Haim, I. Bhattacharya, F. Dinuzzo, A. Saha, N. Slonim, Stance Classification of Context-Dependent Claims, in: M. Lapata, P. Blunsom, A. Koller (Eds.), Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers, Association for

- Computational Linguistics, 2017, pp. 251–261. URL: <https://doi.org/10.18653/v1/e17-1024>.
- [27] A. Bondarenko, M. Fröbe, V. Kasturia, M. Hagen, M. Völske, B. Stein, Webis at TREC 2019: Decision Track, in: E. M. Voorhees, A. Ellis (Eds.), Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019, Gaithersburg, Maryland, USA, November 13-15, 2019, volume 1250 of *NIST Special Publication*, National Institute of Standards and Technology (NIST), 2019. URL: <https://trec.nist.gov/pubs/trec28/papers/Webis.D.pdf>.
- [28] A. Bondarenko, M. Fröbe, J. H. Reimer, B. Stein, M. Völske, M. Hagen, Axiomatic Retrieval Experimentation with `ir_axioms`, in: 45th International ACM Conference on Research and Development in Information Retrieval (SIGIR 2022), ACM, 2022. doi:10.1145/3477495.3531743.
- [29] A. Rana, P. Golchha, R. Juntunen, A. Coajă, A. Elzamarany, C.-C. Hung, S. P. Ponzetto, LeviRANK: Limited Query Expansion with Voting Integration for Document Retrieval and Ranking, in: G. Faggioli, N. Ferro, A. Hanbury, M. Potthast (Eds.), Working Notes Papers of the CLEF 2022 Evaluation Labs, CEUR Workshop Proceedings, 2022.
- [30] M. Aba, M. Azra, M. Gallo, O. Mohammad, I. Piacere, G. Virginio, N. Ferro, SEUPD@CLEF: Team Kueri on Argument Retrieval for Comparative Questions, in: G. Faggioli, N. Ferro, A. Hanbury, M. Potthast (Eds.), Working Notes Papers of the CLEF 2022 Evaluation Labs, CEUR Workshop Proceedings, 2022.