Overview of Touché 2021: Argument Retrieval

Extended Version*

Alexander Bondarenko[1], Lukas Gienapp[2], Maik Fröbe[1], Meriem Beloucif[3], Yamen Ajjour[1], Alexander Panchenko[4], Chris Biemann[3], Benno Stein[5], Henning Wachsmuth[6], Martin Potthast[2] and Matthias Hagen[1]

[1]*Martin-Luther-Universität Halle-Wittenberg*
[2]*Leipzig University*
[3]*Universität Hamburg*
[4]*Skolkovo Institute of Science and Technology*
[5]*Bauhaus-Universität Weimar*
[6]*Paderborn University*
touche@webis.de https://touche.webis.de

Abstract

This paper is a report on the second year of the Touché shared task on argument retrieval held at CLEF 2021. With the goal to provide a collaborative platform for researchers, we organized two tasks: (1) supporting individuals in finding arguments on controversial topics of social importance and (2) supporting individuals with arguments in personal everyday comparison situations.

Unlike in the first year, several of the 27 teams participating in the second year managed to submit approaches that improved upon argumentation-agnostic baselines for the two tasks. Most of the teams made use of last year's Touché data for parameter optimization and fine-tuning their best configurations.

Keywords

Argument retrieval, Controversial questions, Comparative questions, Shared task

1. Introduction

Informed decision making and opinion formation are natural routine tasks. Generally, both of these tasks often involve weighing two or more options. Any choice to be made may be based on personal prior knowledge and experience, but they may also often require searching and processing new knowledge. With the ubiquitous access to various kinds of information on the web—from facts over opinions and anecdotes to arguments—everybody has the chance to acquire knowledge for decision making or opinion formation on almost any topic. However, large amounts of easily accessible information imply challenges such as the need to assess their relevance to the specific topic of interest and to estimate how well an implied stance is justified; no matter whether it is about topics of social importance or "just" about personal decisions. In the simplest form, such a justification might be a collection of basic facts and opinions.

*This overview extends that published as part of the CLEF 2021 proceedings [1].

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

CEUR Workshop Proceedings (CEUR-WS.org)

More complex justifications are often grounded in argumentation, though; for instance, a complex relational aggregation of assertions and evidence pro or con either side, where different assertions or evidential statements support or refute each other.

Furthermore, while web resources such as blogs, community question answering sites, news articles, or social platforms contain an immense variety of opinions and argumentative texts, a notable proportion of these may be of biased, faked, or populist nature. This has motivated argument retrieval research to focus not only on the relevance of arguments, but also on the aspect of their quality. While conventional web search engines support the retrieval of factual information fairly well, they hardly address the deeper analysis and processing of argumentative texts, in terms of mining argument units from these texts, assessing the quality of the arguments, or classifying their stance. To address this, the argument search engine args.me [2] was developed to retrieve arguments relevant to a given controversial topic and to account for the pro or con stance of individual arguments in the result presentation. So far, however, it is limited to a document collection crawled from a few online debate portals, and largely disregards quality aspects. Other argument retrieval systems such as ArgumenText [3] and TARGER [4] take advantage of the large web document collection Common Crawl, but their ability to reliably retrieve arguments to support sides in a decision process is limited. The comparative argumentation machine CAM [5], a system for argument retrieval in comparative search, tries to support decision making in comparison scenarios based on billions of individual sentences from the Common Crawl. Still, it lacks a proper ranking of diverse longer argumentative texts.

To foster research on argument retrieval and to establish an exchange of ideas and datasets among researchers, we organized the second Touché lab on argument retrieval at CLEF 2021.[1] Touché is a collaborative platform[2] to develop and share retrieval approaches that aim to support decisions at a societal level (e.g., "Should hate speech be penalized more, and why?") and at a personal level (e.g., "Should I major in philosophy or psychology, and why?"), respectively. The second year of the Touché lab featured two tasks:

1. Argument retrieval for *controversial* questions from a focused collection of debates to support opinion formation on topics of social importance.
2. Argument retrieval for *comparative* questions from a generic web crawl to support informed decision making.

Approaches to these two tasks that take argumentative quality into account besides topical relevance will help search engines to deliver more accurate argumentative results. Additionally, they will also be an important part of open-domain conversational agents that "discuss" controversial societal topics with humans—as showcased by IBM's Project Debater [6, 7, 8].[3]

The teams that participated in the second year of Touché were able to use the topics and relevance judgments from the first year to develop their approaches. Many trained and optimized learning-based rankers as part of their retrieval pipelines and employed a large variety of preprocessing methods (e.g., stemming, duplicate removal, query expansion), argument quality

[1] The name of the lab is inspired by the usage of the term 'touché' as an exclamation "used to admit that someone has made a good point against you in an argument or discussion." [https://dictionary.cambridge.org/dictionary/english/touche]

[2] https://touche.webis.de/

[3] https://www.research.ibm.com/artificial-intelligence/project-debater/

features, or comparative features (e.g., credibility, part-of-speech tags). Overall, different to the first Touché lab, the majority of the submitted approaches improved over the argumentation-agnostic DirichletLM and BM25F-based baselines. In this paper, we review the participants' approaches in depth and cover all runs in the evaluation results.

2. Previous Work

Queries in argument retrieval often are phrases that describe a controversial topic, questions that ask to compare two options, or even complete arguments themselves [9]. In the Touché lab, we address the first two types in two different shared tasks. Here, we briefly summarize the related work on argument retrieval and on retrieval in comparative scenarios.

2.1. Argument Retrieval

Argument retrieval aims for delivering arguments to support users in making a decision or to help persuading an audience of a specific point of view. An argument is usually modeled as a conclusion with supporting or attacking premises [2]. While a conclusion is a statement that can be accepted or rejected, a premise is a more grounded statement (e.g., a statistical evidence).

The development of an argument search engine is faced with challenges that range from mining arguments from unstructured text to assessing their relevance and quality [2]. Argument retrieval follows several paradigms that start from different sources and perform argument mining and retrieval tasks in different orders [10]. Wachsmuth et al. [2], for instance, extract arguments offline using heuristics that are tailored to online debate portals. Their argument search engine args.me uses BM25F to rank the indexed arguments while giving conclusions more weight than premises. Also Levy et al. [11] use distant supervision to mine arguments offline for a set of topics from Wikipedia before ranking them. Following a different paradigm, Stab et al. [3] retrieve documents from the Common Crawl[4] in an online fashion (no prior offline argument mining) and use a topic-dependent neural network to extract arguments from the retrieved documents at query time. With the two Touché tasks, we address the paradigms of Wachsmuth et al. [2] (Task 1) and Stab et al. [3] (Task 2), respectively.

Argument retrieval should rank arguments according to their topical relevance but also to their quality. What makes a good argument has been studied since the time of Aristotle [12]. Recently, Wachsmuth et al. [13] categorized the different aspects of argument quality into a taxonomy that covers three dimensions: logic, rhetoric, and dialectic. Logic concerns the local structure of an argument, i.e, the conclusion and the premises and their relations. Rhetoric covers the effectiveness of the argument in persuading an audience with its conclusion. Dialectic addresses the relations of an argument to other arguments on the topic. For example, an argument that has many attacking premises might be rather vulnerable in a debate. The relevance of an argument to a query's topic is categorized by Wachsmuth et al. [13] under dialectic quality.

Researchers assess argument relevance by measuring an argument's similarity to a query's topic or incorporating its support/attack relations to other arguments. Potthast et al. [14] evaluate four standard retrieval models at ranking arguments with regard to the quality dimensions

[4] http://commoncrawl.org

of relevance, logic, rhetoric, and dialectic. One of the main findings is that DirichletLM is better at ranking arguments than BM25, DPH, and TF-IDF. Gienapp et al. [15] extend this work by proposing a pairwise strategy that reduces the costs of crowdsourcing argument retrieval annotations in a pairwise fashion by 93% (i.e., annotating only a small subset of argument pairs).

Wachsmuth et al. [16] create a graph of arguments by connecting two arguments when one uses the other's conclusion as a premise. Later on, they exploit this structure to rank the arguments in the graph using PageRank scores [17]. This method is shown to outperform several baselines that only consider the content of the argument and its local structure (conclusion and premises). Dumani et al. [18] introduce a probabilistic framework that operates on semantically similar claims and premises. The framework utilizes support/attack relations between clusters of premises and claims and between clusters of claims and a query. It is found to outperform BM25 in ranking arguments. Later, Dumani and Schenkel [19] also proposed an extension of the framework to include the quality of a premise as a probability by using the fraction of premises that are worse with regard to the three quality dimensions of cogency, reasonableness, and effectiveness. Using a pairwise quality estimator trained on the Dagstuhl-15512 ArgQuality Corpus [20], their probabilistic framework with the argument quality component outperformed the one without it on the 50 Task 1 topics of Touché 2020.

2.2. Retrieval for Comparisons

Comparative information needs in web search have first been addressed by basic interfaces where two to-be-compared products are entered separately in a left and a right search box [21, 22]. Comparative sentences are then identified and mined from product reviews in favor or against one or the other to-be-compared option using opinion mining approaches [23, 24, 25]. Recently, the identification of the comparison preference (the "winning" option) in comparative sentences has been tackled in a more open domain (not just product reviews) by applying feature-based and neural classifiers [26, 27]. Such preference classification forms the basis of the comparative argumentation machine CAM [5] that takes two comparison objects and some comparison aspect(s) as input, retrieves comparative sentences in favor of one or the other object using BM25, and then classifies the sentences' preferences for a final merged result table presentation. A proper argument ranking, however, is still missing in CAM. Chekalina et al. [28] later extended the system to accept comparative questions as input and to return a natural language answer to the user. A comparative question is parsed by identifying the comparison objects, aspect(s), and predicate. The system's answer is either generated directly based on Transformers [29] or by retrieval from an index of comparative sentences.

3. Lab Overview and Statistics

The second edition of the Touché lab received 36 registrations (compared to 28 registrations in the first year), with a majority coming from Germany and Italy, but also from the Americas, Europe, Africa, and Asia (16 from Germany, 10 from Italy, 2 from the United States and Mexico, and 1 each from Canada, India, the Netherlands, Nigeria, the Russian Federation, and Tunisia). Aligned with the lab's fencing-related title, the participants were asked to select a real or fictional swordsman character (e.g., Zorro) as their team name upon registration.

We received result submissions from 27 of the 36 registered teams (up from 17 active teams in the first year). As in the previous edition of Touché, we paid attention to foster the reproducibility of the developed approaches by using the TIRA platform [30] that allows easy software submission and automatic evaluation. Upon registration, each team received an invitation to TIRA to deploy actual software implementations of their approaches. TIRA is an integrated cloud-based evaluation-as-a-service research architecture on which participants can install their software within a dedicated virtual machine. By default, the virtual machines operate the server version of Ubuntu 20.04 with one CPU (Intel Xeon E5-2620), 4 GB of RAM, and 16 GB HDD, but we adjusted the resources to the participants' requirements when needed (e.g., one team asked for 30 GB of RAM, 3 CPUs, and 30 GB of HDD). The participants had full administrative access to their virtual machines. Still, we pre-installed the latest versions of reasonable standard software (e.g., Docker and Python) to simplify the deployment of the approaches.

Using TIRA, the teams could create result submissions via a click in the web UI that then initiated the following pipeline: the respective virtual machine is shut down, disconnected from the internet, and powered on again in a sandbox mode, mounting the test datasets for the respective Touché tasks, and running a team's deployed approach. The interruption of the internet connection ensures that the participants' software works without external web services that may disappear or become incompatible—possible causes of reproducibility issues—but it also means that downloading additional external code or models during the execution was not possible. We offered our support when this connection interruption caused problems during the deployment, for instance, with spaCy that tries to download models if they are not already available on the machine, or with PyTerrier that, in its default configuration, checks for online updates. To simplify participation of teams that do not want to develop a fully-fledged retrieval pipeline on their end, we enabled two exceptions from the interruption of the internet connection for all participants: the APIs of args.me and ChatNoir were available even in the sandbox mode to allow accessing a baseline system for each of the tasks. The virtual machines that the participants used for their submissions will be archived such that the respective systems can be re-evaluated or applied to new datasets as long as the APIs of ChatNoir and args.me remain available—which are both maintained by us.

When a software submission in TIRA really was not possible for some reason, the participants could also simply submit plain run files with their result rankings—an option chosen by 5 of the 27 participating teams. Per task, we allowed each team to submit up to 5 runs whose output must follow the standard TREC-style format.[5] We checked the validity of all submitted run files and of the run files produced via TIRA, asking participants to resubmit their files or to rerun their software in case of validity issues—again, also offering our support in case of problems. All 27 active teams managed to submit at least one valid run. The total of 88 valid runs more than doubles the 41 valid runs from the first year.

4. Task 1: Argument Retrieval for Controversial Questions

The goal of the Touché 2021 lab's first task was to advance technologies that support individuals in forming opinions on socially important controversial topics such as: "Should hate speech

[5] The expected format was described at the lab's web page [https://touche.webis.de].

Table 1
Example topic for Task 1: Argument Retrieval for Controversial Questions.

Number	89
Title	Should hate speech be penalized more?
Description	Given the increasing amount of online hate speech, a user questions the necessity and legitimacy of taking legislative action to punish or inhibit hate speech.
Narrative	Highly relevant arguments include those that take a stance in favor of or opposed to stronger legislation and penalization of hate speech and that offer valid reasons for either stance. Relevant arguments talk about the prevalence and impact of hate speech, but may not mention legal aspects. Irrelevant arguments are the ones that are concerned with offensive language that is not directed towards a group or individuals on the basis of their membership in the group.

be penalized more?”. For such topics, the task was to retrieve relevant and high-quality argumentative texts from the args.me corpus [10], a focused crawl of online debate portals. In this scenario, relevant arguments should help users to form an opinion on the topic and to find arguments that are potentially useful in debates or discussions.

The results of last year’s Task 1 participants indicated that improving upon the “classic” argument-agnostic DirichletLM retrieval model is challenging, but, at the same time, the results of this baseline still left some room for potential improvements. Also, the detection of the degree of argumentativeness and the assessment of the quality of an argument were not “solved” in the first year, but identified as potentially interesting contributions of submissions to the task’s second edition.

4.1. Task Definition

Given a controversial topic formulated as a question, approaches to Task 1 needed to retrieve relevant and high-quality arguments from the args.me corpus, which covers a wide range of timely controversial topics. To enable approaches that leverage training and fine-tuning, the topics and relevance judgments from the 2020 edition of Task 1 were provided.

4.2. Data Description

Topics. We formulated 50 new search questions on controversial topics. Table 1 shows an example consisting of a title (i.e., a question on a controversial topic), a description that summarizes the particular information need and search scenario, and a narrative describing what makes a retrieved result relevant (meant as a guideline for human assessors). We carefully selected the topics by clustering the debate titles in the args.me corpus, formulating questions for a balanced mix of frequent and niche topics—manually ensuring that at least some relevant arguments are contained in the args.me corpus for each topic.

Document Collection. The document collection for Task 1 was the args.me corpus [10]; freely available for download[6] and also accessible via the args.me API.[7] The corpus contains about 400,000 structured arguments crawled from several debate portals (debatewise.org, idebate.org, debatepedia.org, and debate.org), each with a conclusion (claim) and one or more supporting or attacking premises (reasons).

4.3. Judgment Process

The teams' result rankings should be formatted in the "standard" TREC format where document IDs are sorted by descending relevance score for each search topic. Prior to creating the assessment pools, we ran a near-duplicate detection for all submitted runs using the CopyCat framework [31], since near-duplicates might impact evaluation results [32, 33]. The framework found only 1.1% of the arguments in the top-5 results to be near-duplicates (mostly due to debate portal users reusing their arguments in multiple debate threads). We created duplicate-free versions of each result list by removing the documents for which a higher-ranked document is a near-duplicate; in such cases, the next ranked non-near-duplicate then just moved up the ranked list. The top-5 results of the original and the deduplicated runs then formed the judgment pool—created with TrecTools [34]—resulting in 3,711 unique documents that were manually assessed with respect to their relevance and their argumentative quality.

For the assessment, we used the Doccano tool [35] and followed previously suggested annotation guidelines [15, 14]. Our eight graduate and undergraduate student volunteers (all with a computer science background) assessed each argument's relevance to the given topic with four labels (0: not relevant, 1: relevant, 2: highly relevant, or -2: spam) and the argument's rhetorical quality [20] with three labels (0: low quality, 1: sufficient quality, and 2: high quality). To calibrate the annotators' interpretations of the guidelines (i.e., the topics including the narratives and instructions on argument quality), we conducted an initial kappa test in which each annotator had to label the same 15 arguments from 3 topics (5 arguments from each topic). The observed Fleiss' κ values of 0.50 for argument relevance (moderate agreement) and of 0.39 for argument quality (fair agreement) are similar to previous studies [15, 36, 20]. However, we still had a follow-up discussion with all the annotators to clarify potential misinterpretations. Afterwards, each annotator independently judged the results for disjoint subsets of the topics (i.e., each topic was judged by one annotator only).

4.4. Submitted Approaches and Results

Twenty-one participating teams submitted at least one valid run to Task 1. The submissions partly continued the trend of Touché 2020 [37] by deploying "classical" retrieval models, however, with an increased focus on machine learning models (especially for query expansion and for assessing argument quality). Overall, we observed two kinds of contributions: (1) Reproducing and fine-tuning approaches from the previous year by increasing their robustness, and (2) developing new, mostly neural approaches for argument retrieval by fine-tuning pre-trained models for the domain-specific search task at hand.

[6] https://webis.de/data.html#args-me-corpus
[7] https://www.args.me/api-en.html

Like in the first year, combining "classical" retrieval models with various query expansion methods and domain-specific re-ranking features remained a frequent choice of approaches to Task 1. Not really surprising—given last year's baseline results—DirichletLM was employed most often as the initial retrieval model, followed by BM25. For query expansion, most participating teams continued to leverage WordNet [38]. However, transformer-based approaches received increased attention, such as query hallucination, which was successfully used by Akiki and Potthast [39] in the previous Touché lab. Similarly, utilizing deep semantic phrase embeddings to calculate the semantic similarity between a query and possible result documents gained widespread adoption. Moreover, many approaches tried to use some form of argument quality estimation as one of their features for ranking or re-ranking.

This year's approaches benefited from the judgments released for Touché in 2020. Many teams used them for general parameter optimization but also to evaluate intermediate results of their approaches and to fine-tune or select the best configurations. For instance, comparing different kinds of pre-processing methods based on the available judgments from last year received much attention (e.g., stopword lists, stemming algorithms, or duplicate removal).

The results of the runs with the best nDCG@5 scores per participating team are reported in Table 2 (cf. Appendix A for evaluation results of all submitted runs). Below, we review the participants' approaches submitted to Task 1, ordered alphabetically by team name[8]

Asterix [40] preprocesses the args.me corpus by removing duplicate documents and filtering out documents that are too short. The resulting dataset is indexed using BM25. Then a linear regression model on the Webis-ArgQuality-20 argument quality dataset [15] is trained, predicting a given argument's overall quality. At retrieval time, the topic query is expanded using WordNet-based query expansion, 1,000 documents are retrieved using the BM25 index, and then re-ranked using a weighted combination of the normalized predicted quality score and the normalized BM25 score. They optimize the weighting against nDCG@5 using the relevance judgments from Touché 2020. A total of five runs were submitted.

Athos uses a DirichletLM retrieval model with a μ value of 2,000 and indexes the fields of an argument (conclusion and premise) separately. Both fields get preprocessed by lower-casing and removing stop words, urls, and emails. The ranking scores for both fields are then weighted as follows: 0.1 for conclusion and 0.9 for premise. A single run was submitted.

Blade uses a DirichletLM retrieval model in one run, and two variations of a BM25-based retrieval in two further runs. Unfortunately, no further details have been provided.

Batman [41] sets out to quantify the contributions of various steps of a retrieval pipeline, using argument retrieval as their proving ground. A finite search space is defined and effectiveness is systematically measured as more modules are added to the retrieval pipeline. Using relevance judgments from Touché 2020, the best combination of similarity function and tokenizer is determined, and then, gradually, different modules are added, valuate, and frozen, such as different stop word lists, different stemmers, and different filtering approaches. This amounts to a comprehensive grid search in hyperparameter space that allowed for choosing better-working components over worse ones for the retrieval pipeline, and provided for a good comparative overview of them. A total of three runs were submitted.

[8] Nine teams participated in Task 1 with valid runs, but did not submit a notebook describing their approach. Their methodology is summarized in short here, after consulting with the respective team members. *Blade* and *Palpatine* did not provide further information.

Table 2

Results for Task 1: Argument Retrieval for Controversial Questions. The left part (a) shows the evaluation results of a team's best run according to the results' relevance, while the right part (b) shows the best runs according to the results' quality. An asterisk (*) indicates that the runs with the best relevance and the best quality differ for a team. The baseline DirichletLM ranking is shown in bold.

(a) Best relevance score per team

Team	**nDCG@5**	
	Relevance	Quality
Elrond*	0.720	0.809
Pippin Took*	0.705	0.798
Robin Hood*	0.691	0.756
Asterix*	0.681	0.802
Dread Pirate Roberts*	0.678	0.804
Skeletor*	0.667	0.815
Luke Skywalker	0.662	0.808
Shanks*	0.658	0.790
Heimdall*	0.648	0.833
Athos	0.637	0.802
Goemon Ishikawa	0.635	0.812
Jean Pierre Polnareff	0.633	0.802
Swordsman	**0.626**	**0.796**
Yeagerists	0.625	0.810
Hua Mulan*	0.620	0.789
Macbeth*	0.611	0.783
Blade*	0.601	0.751
Deadpool	0.557	0.679
Batman	0.528	0.695
Little Foot	0.521	0.718
Gandalf	0.486	0.603
Palpatine	0.401	0.562

(b) Best quality score per team

Team	**nDCG@5**	
	Quality	Relevance
Heimdall*	0.841	0.639
Skeletor*	0.827	0.666
Asterix*	0.818	0.663
Elrond*	0.817	0.674
Pippin Took*	0.814	0.683
Goemon Ishikawa	0.812	0.635
Hua Mulan*	0.811	0.620
Dread Pirate Roberts*	0.810	0.647
Yeagerists	0.810	0.625
Robin Hood*	0.809	0.641
Luke Skywalker	0.808	0.662
Macbeth*	0.803	0.608
Athos	0.802	0.637
Jean Pierre Polnareff	0.802	0.633
Swordsman	**0.796**	**0.626**
Shanks*	0.795	0.639
Blade*	0.763	0.588
Little Foot	0.718	0.521
Batman	0.695	0.528
Deadpool	0.679	0.557
Gandalf	0.603	0.486
Palpatine	0.562	0.401

Deadpool applies a query expansion technique with a DirichletLM model (μ=4000). Both the conclusion and the premise of an argument are indexed, with 0.1 and 0.9 weights, respectively. The query expansion technique relies on the top-5 arguments to derive terms that associated with the query term. To quantify the co-occurrence of a term in an argument with the query terms, its conditional probability to that of the query terms are calculated and smoothed by the term's inverse document frequency. The conditional probability of a term given a query term is calculated using the count of arguments that contain both terms, divided by the count of arguments that contains the query term. A single run was submitted.

Dread Pirate Roberts [42] uses four classes of approaches to retrieve relevant arguments from the args.me corpus for a query on a controversial topic. Therefore, *Roberts* contrasts two "traditional" approaches with two novel approaches. The traditional approaches involve one run that uses a Dirichlet-smoothed language-model with low-quality arguments removed by argument clustering with the Universal Sentence Encoder model [43], and two feature-based learning to rank approaches with LambdaMART [44]. The learning to rank models are trained on the relevance labels of Task 1 of Touché 2020 and differ in the used features. With 31 features belonging to 5 different feature classes as starting point, *Roberts* runs a greedy feature-selection identifying a subset of 4 and 9 features with best nDCG scores in a five-fold cross-validation setup. Afterwards, both feature sets are used on all relevance labels of Task 1 of Touché 2020 to train dedicated LambdaMART models that re-rank the top-100 results of the DirichletLM retrieval, producing 2 LambdaMART runs. *Roberts* further submits one run that re-ranks the top-100 results of the DirichletLM retrieval with a question-answering model. The idea behind this run is to phrase the task to retrieve relevant arguments for a controversial query as deciding whether an argument "answers" the controversial query. Therefore, the question-answering retrieval model coming with the Universal Sentence Encoder scores the top-100 argument for a query whether the argument "answers" the query or not, sorting the arguments by descending question-answering score. The fifth run submitted by Dread Pirate Roberts uses transformer-based query expansion where the query is expanded with keywords generated with RoBERTa [45]. Therefore, Dread Pirate Roberts embedded the controversial query into a pattern letting RoBERTa predict tokens, expanding the query with the top-10 tokens and their RoBERTa score as a weighted query submitted to the DirichletLM retrieval model. A total of five runs were submitted.

Elrond focuses on implementing a document analyzing pipeline to be used together with a DirichletLM-based retrieval. They rely on the Krovetz stemming algorithm and remove stop words using a custom stop list. They also compute part-of-speech tags and remove tokens from documents by filtering out certain tags. Documents are further enriched using WordNet-based synonyms. A total of four runs were submitted.

Gandalf indexes for each argument only the conclusion and uses BM25 as a retrieval model in a single-run submission.

Goemon Ishikawa [46] explores different configurations of a standard Lucene-based retrieval pipeline, varying the similarity function (BM25, DirichletLM), tokenizers (Lucene, OpenNLP), stop word lists (Lucene, Atire, Terrier, Smart), and lemmatizers (OpenNLP). Additionally, they test query expansion with synonyms from WordNet. Thirteen such configurations were evaluated on topics from Touché 2020 with respect to average precision, precision@10, nDCG, and nDCG@5. In an analysis of variance, they observe overall high variances for all evaluation measures and configurations, and that DirichletLM-based configurations perform significantly better, however, the effect of different tokenizers, stop word lists, or lemmatizers could not be assessed conclusively. A manual analysis by the authors on two topics suggests that expanding the query with synonyms can possibly drift the query. Using five DirichletLM models, two of which expand the query, and non apply lemmatization, a total of five runs were submitted.

Heimdall [47] aims at including both topical relevance and argument quality while ranking arguments. As a basic retrieval model, DirichletLM is used. The basic retrieval model is considered to give a mere textual relevance. To assess the topical relevance of an argument, arguments are embedded using the Universal Sentence Encoder and then clustered using k-means with $k = 300$. Arguments are then represented using their cluster centroids and the topical relevance of an argument is calculated using the cosine similarity of the query to the centroid. Argument quality is assessed using a support vector regression model that is trained on the Webis-ArgQuality-20 corpus. The regression model achieves a mean squared error of 0.19. Before assessing the quality of arguments, an argumentativeness classifier is used to filter input instances that are not arguments. The support vector machine classifier is also trained on the same dataset and achieves an F1-score of 0.88. A total of five runs were submitted.

Hua Mulan [48] proposes to expand documents from the args.me corpus prior to indexing, evaluating how different expansion methods affect the argument retrieval for controversial topics. Three expansion approaches are presented: the first uses a transformer-based query prediction to generate queries based on the premises and conclusions as input, which are then added to the documents. The second is also transformer-based and generates ("hallucinates") arguments using GPT-2 based on the conclusions. The third approach uses TF-IDF to determine the top-10 keywords and expands the premises using synonyms from the WordNet database. For evaluation, all corpora were indexed and retrieved using Elasticsearch and the DirichletLM similarity. The altered args.me corpus with expansions is made available as dataset. A total of three runs were submitted.

Jean-Pierre Polnareff [49] combines differently weighted versions of the BM25 and DirichletLM retrieval model with a WordNet-based query expansion, and a re-ranking component that incorporates sentiment analysis to explore whether boosting arguments with high sentiment scores or boosting neutral arguments leads to better results. The authors provide an ablative evaluation study for each of these three components, motivating their parameter choice at each step. Furthermore, different text pre-processing steps were reviewed in-depth, evaluating the effect of the choice of stop word list and stemming algorithm on the final result. A single run was submitted.

Little Foot applies a query expansion technique over an Okapi BM25 model. The team indexes three fields for each argument: conclusion, premise, and context. Preprocessing the three fields includes lemmatization and removing stop words. The query expansion technique expands nouns, adjectives, and adverbs in the query with synonyms from WordNet. When multiple meanings exist for a word (known as "synset" in WordNet jargon), the approach uses the Lesk algorithm [50] to disambiguate the meaning of the word based on the context. A single run was submitted.

Luke Skywalker indexes for each argument its premise, conclusion, and context. As a retrieval model they implemented their own $tf \cdot idf$ model in a single-run submission.

Macbeth [51] describes an approach that utilizes fine-tuned SBERT sentence embeddings [52] in conjunction with different retrieval strategies. First, further pre-training of the RoBERTa model on the args.me corpus with annotated relevance labels is carried out. They then obtain sentence embeddings of all documents in the args.me corpus with SBERT based on the pre-trained model. Weakly supervised data-augmentation is used to fine-tune the bi-encoder further, based on labels inferred using a cross-encoder architecture. Three retrieval strategies are then applied: (1) approximate nearest-neighbor vector retrieval on the inferred document embeddings, (2) BM25, and (3) a mixture of both. An initial retrieved pool of candidate documents is re-ranked by direct query/document comparison using a cross-encoder architecture. The authors experiment with different pipeline configurations. A total of five runs were submitted.

Palpatine, befittingly, submitted one of the worst-performing of all runs, without providing any explanation whatsoever.

Pippin Took [53] first preprocesses documents with the Krovetz Stemmer [54], and remove stop words using a custom stop word list curated from various libraries. After parameter-tuning Lucene's implementation of DirichletLM using the Touché 2020 relevance labels, they then experiment with two different retrieval pipelines: (1) query expansion with WordNet, and (2) phrase search with term trigrams, which follows the idea that arguments containing parts of the query as phrases will be part of an effective argumentative ranking. Therefore, the arguments are indexed as term trigrams, and each query is split into term trigrams to retrieve arguments with DirichletLM. However, preliminary experiments suggested that argument retrieval with term trigrams substantially decreases the nDCG@5. Hence, *Took* omits phrase search and submits three runs with DirichletLM only, and two runs with DirichletLM and query expansion, varying the parameter μ of DirichletLM, for a total of five runs.

Robin Hood relies on the RM3 implementation from the Pyserini toolkit [55] to perform query expansion. For retrieval, they embed both the premise and the conclusion of each argument into two separate vector spaces using the Universal Sentence Encoder, ranking arguments based on the cosine similarity between embedded query and document. The two embeddings are incorporated with different weights. They further take document length into account, deducting up to 15% of an arguments score if its length lies outside of one standard deviation of the mean across the whole corpus. They submit one baseline run using the DirichletLM retrieval model, one with RM3 query expansion applied on top of that, one using only cosine similarity on phrase embeddings, and one using RM3 in conjunction with phrase embeddings for retrieval, for a total of four runs.

Shanks [56] indexes discussion titles in addition to the premises and conclusions in the args.me corpus. They construct a custom stop word list based on the Smart and Lucene lists, as well as frequent terms within the document collection. They then use a Boolean model with the individual terms of the query to apply boosts to the indexed documents. Each matched term between query and discussion titles, conclusions, and premises in the corpus, as well as all identified WordNet synonyms of query terms receive a boosting factor. Both BM25 and DirichletLM are then used to retrieve relevant documents, with boosting applied. Additionally, a proximity search for all term pairs within the query can be performed and boosted individually. A total of five runs were submitted.

Skeletor [57] submits five runs using three different approaches: (1) BM25 retrieval, (2) ranking arguments based on their semantic similarity to the query, and (3) using pseudo relevance feedback in combination with the semantic similarity of passages. Unanimously, the arguments' premise is used for ranking. The BM25 approach uses Pyserini with the BM25 parameters k_1 and b fine-tuned with grid search on the relevance judgments from Touché 2020. The two semantic similarity runs use the model msmarco-distilbert-base-v3 provided by Sentence Transformers [52], which was fine-tuned for question-answering on MS MARCO [58]. Therefore, arguments are split by sentence into passages of approximately 200 words, using the maximum cosine similarity of all passages in the argument to the encoded query as retrieval score. The submitted runs differ as follows: Run 1 ranks documents solely by their semantic similarity to the query using approximate nearest neighbor search; Runs 2 and 3 interpolate the semantic similarity score with the tuned BM25 scores; Runs 4 and 5 use the top-3 arguments retrieved by the interpolation of BM25 with the semantic similarity score as pseudo relevance feedback: for each passage from the relevance feedback, the 50 most similar passages are identified with an approximate nearest neighbor search on all encoded passages of the corpus. The probabilities that a passage is highly similar to a passage in the pseudo relevance feedback are determined with manifold approximation and summed as the argument's score. In Run 4 all arguments in the corpus are ranked with this score, and in Run 5 only the top-10 results of the interpolation of BM25 with the semantic similarity are re-ranked.

The baseline run of *Swordsman* encompasses two separate approaches: the Elasticsearch implementation of query likelihood with Dirichlet-smoothed language models (DirichletLM [59]), as well as the args.me API.

The *Yeagerists* [60] describe an approach that integrates two components: query expansion and argument quality regression. Query expansion is performed using a pretrained BERT model which is prompted to substitute certain masked words (adjectives, nouns, and past participles) in the topics. Argument quality regression is performed by training a BERT as a regression model on Webis-ArgQuality-20. The regression model is trained in a 8:1:1 split using mean squared error (MSE) as a loss function, and achieves an MSE of 0.728 on the test split. At retrieval time, for each topic, ten queries are generated using the lexical substitution algorithm and then forwarded to a DirichletLM retrieval model to produce a relevance score. The top-100 arguments are then passed to the regression model to predict their quality score. The relevance score and quality score are normalized and averaged with a weighting variable α that controls the contribution of the quality score to the averaged score. The team tests different α-values using the relevance labels from Touché 2020 to motivate parameter choices for their submitted runs. A total of five runs were submitted.

5. Task 2: Argument Retrieval for Comparative Questions

The goal of the Touché 2021 lab's second task was to support individuals making informed decisions in "everyday" or personal comparison situations—in its simplest form for questions such as "Is X or Y better for Z?". Decision making in such situations benefits from finding balanced justifications for choosing one or the other option, for instance, via an overview of relevant and high-quality pro/con arguments.

Similar to Task 1, the results of last year's Task 2 participants indicated that improving upon an argument-agnostic BM25F baseline is challenging. Promising proposed approaches tried to re-rank based on features capturing "comparativeness" or "argumentativeness."

5.1. Task Definition

Given a comparative question, an approach to Task 2 needed to retrieve documents from the general web crawl ClueWeb12[9] that help to come to an informed decision on the comparison. Ideally, the retrieved documents should be argumentative with convincing arguments for or against one or the other option. To identify arguments in web documents, the participants were not restricted to any system; they could use own technology or any existing argument taggers such as MARGOT [61]. To lower the entry barriers for participants new to argument mining, we offered support for using the neural argument tagger TARGER [4], hosted on our own servers and accessible via an API.[10]

[9] https://lemurproject.org/clueweb12/

[10] https://demo.webis.de/targer-api/apidocs/

Table 3
Example topic for Task 2: Argument Retrieval for Comparative Questions.

Number	88
Title	Should I major in philosophy or psychology?
Description	A soon-to-be high-school graduate finds themself at a crossroad in their live. Based on their interests, majoring in philosophy or in psychology are the potential options and the graduate is searching for information about the differences and similarities, as well as advantages and disadvantages of majoring in either of them (e.g., with respect to career opportunities or gained skills).
Narrative	Relevant documents will overview one of the two majors in terms of career prospects or developed new skills, or they will provide a list of reasons to major in one or the other. Highly relevant documents will compare the two majors side-by-side and help to decide which should be preferred in what context. Not relevant are study program and university advertisements or general descriptions of the disciplines that do not mention benefits, advantages, or pros/cons.

5.2. Data Description

Topics. For the second edition of Task 2, we manually selected 50 new comparative questions from the MS MARCO dataset [58] (questions from Bing's search logs) and the Quora dataset [62] (questions asked on the Quora question answering website). We ensured to have questions on diverse topics, for example, asking about electronics, cuisine, house appliances, life choices, etc. Table 3 shows an example topic for Task 2 that consists of a title (i.e., a comparative question), a description of the possible search context and situation, and a narrative describing what makes a retrieved result relevant (meant as a guideline for human assessors). In the topic selection, we ensured that relevant documents for each topic were actually contained in the ClueWeb12 (i.e., avoiding questions on comparison options not known at the ClueWeb12 crawling time in 2012).

Document Collection. The document collection was formed by the ClueWeb12 dataset that contains 733 million English web pages (27.3 TB uncompressed), crawled by the Language Technologies Institute at Carnegie Mellon University between February and May 2012. For participants of Task 2 who could not index the ClueWeb12 at their site, we provided access to the indexed corpus through the BM25F-based search engine ChatNoir [63] via its API.[11]

5.3. Judgment Process

Using the CopyCat framework [31], we found that, on average, 11.6% of the documents in the top-5 results of a run were near-duplicates—a non-negligible redundancy that might have negatively impacted the reliability and validity of our evaluation since rankings containing multiple relevant duplicates tend to overestimate the actual retrieval effectiveness [32, 33].

[11] https://www.chatnoir.eu/doc/

Following the strategy used in Task 1, we pooled the top-5 documents from the original and the deduplicated runs, resulting in 2,076 unique documents that needed to be judged.

Our eight volunteer annotators (same as for Task 1) labeled a document for its topical relevance (three labels; 0: not relevant, 1: relevant, and 2: highly relevant) and for whether rhetorically well-written arguments [20] were contained (three labels; 0: low quality or no arguments in the document, 1: sufficient quality, and 2: high quality). Similar to Task 1, our eight volunteer assessors went through an initial kappa test on 15 documents from 3 topics (5 documents per topic). As in case of Task 1, the observed Fleiss' κ values of 0.46 for relevance (moderate agreement) and of 0.22 for quality (fair agreement) are similar to previous studies [15, 36, 20]. Again, however, we had a follow-up discussion with all the annotators to clarify some potential misinterpretations. Afterwards, each annotator independently judged the results for disjoint subsets of the topics (i.e., each topic was judged by one annotator only).

5.4. Submitted Approaches and Results

For Task 2, six teams submitted approaches that all used ChatNoir for an initial document retrieval, either by submitting the original topic titles as queries, or by applying query preprocessing (e.g., lemmatization and POS-tagging) and query expansion techniques (e.g., synonyms from WordNet [38], or generated with word2vec [64] or sense2vec embeddings [65]). On the retrieved ChatNoir results, most teams then applied a document "preprocessing" (e.g., removing HTML markup) before re-ranking with feature-based or neural classifiers trained on last year's judgments with, for instance, argumentativeness, credibility, or comparativeness scores as features. The teams predicted document relevance labels by using a random forest classifier, XGBoost [66], LightGBM [67], or a fine-tuned BERT [29]. The results of the runs with the best nDCG@5 scores per participating team are reported in Table 4 (cf. Appendix A for the evaluation results of all submitted runs). Below, we give an overview of the approaches submitted to Task 2, ordered alphabetically by team name.[12]

Jack Sparrow [68] lemmatizes the question queries in a preprocessing step, creates expansion terms by detecting "comparison" terms in the questions (e.g., nouns or comparative adjectives/adverbs as identified by spaCy's POS tagger[13]), and identifies synonyms of these terms from WordNet synsets [38], from word2vec [64], and sense2vec embeddings [65]. The top-100 ChatNoir results returned for the preprocessed and expanded questions are then re-ranked by a support vector regression trained on the Touché 2020 topics and judgments to predict relevance scores for the documents using combinations of the following normalized features: (1) argumentative score (sum of argumentativeness probabilities returned by TARGER for each token inside premises and claims), (2) (pseudo) trustworthiness score (0–10-valued PageRank scores obtained from Open PageRank)[14], (3) relevance labels predicted by a BERT-based classifier fine-tuned on the Touché 2020 topics and judgments, and (4) the ChatNoir relevance score. Different runs of *Sparrow* use different combinations of query preprocessing and expansion, and different feature combinations for the support vector regression; the most effective run

[12] One team participated in Task 2 with a valid run, but did not submit a notebook describing their approach. Their methodology is summarized in short here, after consulting with the team members.

[13] https://spacy.io/

[14] https://www.domcop.com/openpagerank/what-is-openpagerank

Table 4
Results for Task 2: Argument Retrieval for Comparative Questions. The left part (a) shows the evaluation results of a team's best run according to the results' relevance, while the right part (b) shows the best runs according to the results' quality. An asterisk (⋆) indicates that the runs with the best relevance and the best quality differ for a team. The baseline ChatNoir ranking is shown in bold.

(a) Best relevance score per team

Team	**nDCG@5**	
	Relevance	Quality
Katana⋆	0.489	0.675
Thor	0.478	0.680
Rayla⋆	0.473	0.670
Jack Sparrow	0.467	0.664
Mercutio	0.441	0.651
Puss in Boots	**0.422**	**0.636**
Prince Caspian	0.244	0.548

(b) Best quality score per team

Team	**nDCG@5**	
	Quality	Relevance
Rayla⋆	0.688	0.466
Katana⋆	0.684	0.460
Thor	0.680	0.478
Jack Sparrow	0.664	0.467
Mercutio	0.651	0.441
Puss in Boots	**0.636**	**0.422**
Prince Caspian	0.548	0.244

uses query lemmatization and expansion while the regression is trained on the BERT relevance predictions, combined with the ChatNoir relevance scores. A total of four runs were submitted.

Katana [69] re-ranks the top-100 ChatNoir results (original questions as queries) but using different feature-based and neural classifiers/rankers to predict the final relevance labels: (1) an XGBoost [66] approach (overall relevance-wise most effective run), (2) a LightGBM [67] approach (team Katana's quality-wise best run), (3) Random Forests [70], and (4) a BERT-based ranker from OpenNIR [71]. The feature-based approaches are trained on the topics and judgments from Touché 2020, employing a range of relevance features (e.g., ChatNoir relevance score) and "comparativness" features (e.g., number of identified comparison objects, aspects, or predicates [28]). The BERT-based ranker is trained on the ANTIQUE question-answering dataset [72] (34,000 text passages with relevance annotations for 2,600 open-domain non-factoid questions). A total of six runs were submitted (we evaluated all of them since the overall judgment load was feasible).

Mercutio [73] expands the original question queries with synonyms obtained from word2vec embeddings [64] (*Mercutio*'s best run uses embeddings pre-trained on the Gigaword corpus[15]) or nouns found in GPT-2 [74] extensions when prompted with the question. The respective top-100 ChatNoir results are then re-ranked based on a linear combination of several scores (e.g., term-frequency counts, ratio of premises and claims in documents as identified by TARGER, etc.). The weights of the individual scores are optimized in a grid search on the Touché 2020 topics and judgments. A total of three runs were submitted.

Prince Caspian re-ranks the top-40 ChatNoir results returned for the questions without stop words. The re-ranking uses the results' main content (extracted with the BoilerPy3 library;[16] topic title terms in the extracted main content masked with a "MASK" token) and a logistic regression classifier (features: $tf \cdot idf$-weighted 1- to 4-grams; training on the Touché 2020 topics and judgments) that predicts the probability of a result being relevant (final ranking by

[15] https://catalog.ldc.upenn.edu/LDC2011T07
[16] https://pypi.org/project/boilerpy3/

descending probability). A single run was submitted.

The baseline run of *Puss in Boots* simply uses the results that ChatNoir [63] returns for the original question query. ChatNoir is an Elasticsearch-based search engine for the ClueWeb12 (and several other web corpora) that employs BM25F ranking (fields: document title, keywords, main content, and the full document) and SpamRank scores [75].

Rayla [76] uses two query processing/expansion techniques: (1) removing stop words and punctuation, and then lemmatizing the remaining tokens with spaCy, and (2) expanding comparative adjectives/adverbs (POS-tagged with spaCy) with a maximum of five synonyms and antonyms. The final re-ranking is created by linearly combining different scores such as a ChatNoir's relevance score, PageRank, and SpamRank (both also returned by ChatNoir), an argument support score (ratio of argumentative sentences (premises and claims) in documents found with a custom DistilBERT-based [77] classifier), and a similarity score (averaged cosine similarity between the original query and every argumentative sentence in the document represented by Sentence-BERT embeddings [52]). The weights of the individual scores are optimized in a grid search on the Touché 2020 topics and judgments. A total of four runs were submitted.

Thor [78] removes, as query preprocessing, any punctuation from the topic titles. They then locally create an Elasticsearch BM25F index of the top-110 ChatNoir results (fields: original and lemmatized document title, document body extracted using the BoilerPy3 library, and premises and claims as identified by TARGER in the body) with the BM25 parameters optimized by a grid search on the Touché 2020 judgments ($b = 0.68$ and $k_1 = 1.2$). The local index is then queried with the lemmatized topic title expanded by WordNet synonyms [38]. A single run was submitted.

6. Summary and Outlook

From the 36 teams that registered for the Touché 2021 lab, 27 actively participated by submitting at least one valid run to one of the two shared tasks: (1) argument retrieval for controversial questions, and (2) argument retrieval for comparative questions. Most of the participating teams used the judgments from the first lab's edition to train feature-based or neural approaches that predict argument quality or that re-rank some initial retrieval result set. Overall, many more approaches could improve upon the argumentation-agnostic baselines (DirichletLM for Task 1 and BM25F for Task 2) than in the first year, indicating that progress was achieved. For a potential third year of the Touché lab, we currently plan to focus on retrieving the most relevant/argumentative text passages and on detecting the pro/con stance of the returned results.

Acknowledgments

We are very grateful to the CLEF 2021 organizers and the Touché participants, who allowed this lab to happen. We also want to thank Jan Heinrich Reimer for setting up Doccano, Christopher Akiki for providing the baseline DirichletLM implementation *Swordsman*, our volunteer annotators who helped to create the relevance and argument quality assessments, and our reviewers for their valuable feedback on the participants' notebooks.

This work was partially supported by the DFG through the project “ACQuA: Answering Comparative Questions with Arguments” (grants BI 1544/7-1 and HA 5851/2-1) as part of the priority program “RATIO: Robust Argumentation Machines” (SPP 1999).

References

[1] A. Bondarenko, L. Gienapp, M. Fröbe, M. Beloucif, Y. Ajjour, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, M. Potthast, M. Hagen, Overview of Touché 2021: Argument Retrieval, in: Proceedings of the 12th International Conference of the CLEF Association (CLEF 2021), volume 12880 of *Lecture Notes in Computer Science*, Springer, 2021.

[2] H. Wachsmuth, M. Potthast, K. A. 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 (ArgMining@EMNLP 2017), Association for Computational Linguistics, 2017, pp. 49–59. URL: https://doi.org/10.18653/v1/w17-5106.

[3] 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 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2018), Association for Computational Linguistics, 2018, pp. 21–25. URL: https://doi.org/10.18653/v1/n18-5005.

[4] A. Chernodub, O. Oliynyk, P. Heidenreich, A. Bondarenko, M. Hagen, C. Biemann, A. Panchenko, TARGER: Neural Argument Mining at Your Fingertips, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019), Association for Computational Linguistics, 2019, pp. 195–200. URL: https://www.aclweb.org/anthology/P19-3031.

[5] M. Schildwächter, A. Bondarenko, J. Zenker, M. Hagen, C. Biemann, A. Panchenko, Answering Comparative Questions: Better than Ten-Blue-Links?, in: Proceedings of the Conference on Human Information Interaction and Retrieval (CHIIR 2019), Association for Computing Machinery, 2019, pp. 361–365. URL: https://doi.org/10.1145/3295750.3298916.

[6] R. Bar-Haim, L. Eden, R. Friedman, Y. Kantor, D. Lahav, N. Slonim, From Arguments to Key Points: Towards Automatic Argument Summarization, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), Association for Computational Linguistics, 2020, pp. 4029–4039. URL: https://doi.org/10.18653/v1/2020.acl-main.371.

[7] R. Bar-Haim, D. Krieger, O. Toledo-Ronen, L. Edelstein, Y. Bilu, A. Halfon, Y. Katz, A. Menczel, R. Aharonov, N. Slonim, From Surrogacy to Adoption; From Bitcoin to Cryptocurrency: Debate Topic Expansion, in: Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL 2019), Association for Computational Linguistics, 2019, pp. 977–990. URL: https://doi.org/10.18653/v1/p19-1094.

[8] Y. Mass, S. Shechtman, M. Mordechay, R. Hoory, O. S. Shalom, G. Lev, D. Konopnicki, Word Emphasis Prediction for Expressive Text to Speech, in: Proceedings of the 19th Annual Conference of the International Speech Communication Association (Interspeech 2018), ISCA, 2018, pp. 2868–2872. URL: https://doi.org/10.21437/Interspeech.2018-1159.

[9] H. Wachsmuth, S. Syed, B. Stein, Retrieval of the Best Counterargument without Prior Topic Knowledge, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018), Association for Computational Linguistics, 2018, pp. 241–251. URL: https://www.aclweb.org/anthology/P18-1023/.

[10] Y. Ajjour, H. Wachsmuth, J. Kiesel, M. Potthast, M. Hagen, B. Stein, Data Acquisition for Argument Search: The args.me Corpus, in: Proceedings of the 42nd German Conference on Artificial Intelligence (KI 2019), Springer, 2019, pp. 48–59. doi:10.1007/978-3-030-30179-8_4.

[11] R. Levy, B. Bogin, S. Gretz, R. Aharonov, N. Slonim, Towards an Argumentative Content Search Engine using Weak Supervision, in: Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018), Association for Computational Linguistics, 2018, pp. 2066–2081. URL: https://www.aclweb.org/anthology/C18-1176/.

[12] Aristotle, G. A. Kennedy, On Rhetoric: A Theory of Civic Discourse, Oxford: Oxford University Press, 2006.

[13] H. Wachsmuth, N. Naderi, I. Habernal, Y. Hou, G. Hirst, I. Gurevych, B. Stein, Argumentation Quality Assessment: Theory vs. Practice, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017), Association for Computational Linguistics, 2017, pp. 250–255. URL: https://doi.org/10.18653/v1/P17-2039.

[14] M. Potthast, L. Gienapp, F. Euchner, N. Heilenkötter, N. Weidmann, H. Wachsmuth, B. Stein, M. Hagen, Argument Search: Assessing Argument Relevance, in: Proceedings of the 42nd International Conference on Research and Development in Information Retrieval (SIGIR 2019), Association for Computing Machinery, 2019, pp. 1117–1120. URL: https://doi.org/10.1145/3331184.3331327.

[15] 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 (ACL 2020), Association for Computational Linguistics, 2020, pp. 5772–5781. URL: https://www.aclweb.org/anthology/2020.acl-main.511/.

[16] 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 (EACL 2017), Association for Computational Linguistics, 2017, pp. 1117–1127. URL: https://doi.org/10.18653/v1/e17-1105.

[17] 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/.

[18] L. Dumani, P. J. Neumann, R. Schenkel, A Framework for Argument Retrieval - Ranking Argument Clusters by Frequency and Specificity, in: Proceedings of the 42nd European Conference on IR Research (ECIR 2020), volume 12035 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 431–445. URL: https://doi.org/10.1007/978-3-030-45439-5_29.

[19] L. Dumani, R. Schenkel, Quality Aware Ranking of Arguments, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM 2020), Association for Computing Machinery, 2020, pp. 335–344. URL: https://doi.org/10.1007/978-3-030-45439-5_29.

[20] 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: Proceedings

of the 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.

[21] A. Nadamoto, K. Tanaka, A Comparative Web Browser (CWB) for Browsing and Comparing Web Pages, in: Proceedings of the 12th International World Wide Web Conference (WWW 2003), Association for Computing Machinery, 2003, pp. 727–735. URL: https://doi.org/10.1145/775152.775254.

[22] J. Sun, X. Wang, D. Shen, H. Zeng, Z. Chen, CWS: A Comparative Web Search System, in: Proceedings of the 15th International Conference on World Wide Web (WWW 2006), Association for Computing Machinery, 2006, pp. 467–476. URL: https://doi.org/10.1145/1135777.1135846.

[23] N. Jindal, B. Liu, Identifying Comparative Sentences in Text Documents, in: Proceedings of the 29th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2006), Association for Computing Machinery, 2006, pp. 244–251. URL: https://doi.org/10.1145/1148170.1148215.

[24] N. Jindal, B. Liu, Mining Comparative Sentences and Relations, in: Proceedings of the 21st National Conference on Artificial Intelligence and the 18th Innovative Applications of Artificial Intelligence Conference (AAAI 2006), AAAI Press, 2006, pp. 1331–1336. URL: http://www.aaai.org/Library/AAAI/2006/aaai06-209.php.

[25] W. Kessler, J. Kuhn, A Corpus of Comparisons in Product Reviews, in: Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014), European Language Resources Association (ELRA), 2014, pp. 2242–2248. URL: http://www.lrec-conf.org/proceedings/lrec2014/summaries/1001.html.

[26] A. Panchenko, A. Bondarenko, M. Franzek, M. Hagen, C. Biemann, Categorizing Comparative Sentences, in: Proceedings of the 6th Workshop on Argument Mining (ArgMining@ACL 2019), Association for Computational Linguistics, 2019, pp. 136–145. URL: https://doi.org/10.18653/v1/w19-4516.

[27] N. Ma, S. Mazumder, H. Wang, B. Liu, Entity-Aware Dependency-Based Deep Graph Attention Network for Comparative Preference Classification, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), Association for Computational Linguistics, 2020, pp. 5782–5788. URL: https://www.aclweb.org/anthology/2020.acl-main.512/.

[28] V. Chekalina, A. Bondarenko, C. Biemann, M. Beloucif, V. Logacheva, A. Panchenko, Which is Better for Deep Learning: Python or MATLAB? Answering Comparative Questions in Natural Language, in: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations (EACL 2021), Association for Computational Linguistics, 2021, pp. 302–311. URL: https://www.aclweb.org/anthology/2021.eacl-demos.36/.

[29] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), Association for Computational Linguistics, 2019, pp. 4171–4186. URL: https://doi.org/10.18653/v1/n19-1423.

[30] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, TIRA Integrated Research Architecture, in: 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.

[31] M. Fröbe, J. Bevendorff, L. Gienapp, M. Völske, B. Stein, M. Potthast, M. Hagen, CopyCat: Near-Duplicates within and between the ClueWeb and the Common Crawl, in: Proceedings of the 44th International ACM Conference on Research and Development in Information Retrieval (SIGIR 2021), Association for Computing Machinery, 2021. URL: https://dl.acm.org/doi/10.1145/3404835.3463246.

[32] M. Fröbe, J. Bevendorff, J. Reimer, M. Potthast, M. Hagen, Sampling Bias Due to Near-Duplicates in Learning to Rank, in: Proceedings of the 43rd International ACM Conference on Research and Development in Information Retrieval (SIGIR 2020), Association for Computing Machinery, 2020, pp. 1997–2000. URL: https://doi.org/10.1145/3397271.3401212.

[33] M. Fröbe, J. Bittner, M. Potthast, M. Hagen, The Effect of Content-Equivalent Near-Duplicates on the Evaluation of Search Engines, in: Proceedings of the 42nd European Conference on IR Research (ECIR 2020), volume 12036 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 12–19. doi:10.1007/978-3-030-45442-5_2.

[34] J. R. M. Palotti, H. Scells, G. Zuccon, TrecTools: an Open-source Python Library for Information Retrieval Practitioners Involved in TREC-like Campaigns, in: Proceedings of the 42nd International Conference on Research and Development in Information Retrieval (SIGIR 2019), Association for Computing Machinery, 2019, pp. 1325–1328. URL: https://doi.org/10.1145/3331184.3331399.

[35] H. Nakayama, T. Kubo, J. Kamura, Y. Taniguchi, X. Liang, Doccano: Text Annotation Tool for Human, 2018. URL: https://github.com/doccano/doccano, software available from https://github.com/doccano/doccano.

[36] H. Wachsmuth, N. Naderi, I. Habernal, Y. Hou, G. Hirst, I. Gurevych, B. Stein, Argumentation Quality Assessment: Theory vs. Practice, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017), Association for Computational Linguistics, 2017, pp. 250–255.

[37] 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: Working Notes Papers of the CLEF 2020 Evaluation Labs, volume 2696 of *CEUR Workshop Proceedings*, 2020. URL: http://ceur-ws.org/Vol-2696/.

[38] C. Fellbaum, WordNet: An Electronic Lexical Database, Bradford Books, 1998.

[39] C. Akiki, M. Potthast, Exploring Argument Retrieval with Transformers, in: Working Notes Papers of the CLEF 2020 Evaluation Labs, volume 2696, 2020. URL: http://ceur-ws.org/Vol-2696/.

[40] E. Raimondi, M. Alessio, N. Levorato, A Search Engine System for Touché Argument Retrieval task to answer Controversial Questions, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

[41] E. Raimondi, M. Alessio, N. Levorato, Step approach to information retrieval., in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

[42] C. Akiki, M. Fröbe, M. Hagen, M. Potthast, Learning to Rank Arguments with Feature Selection, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum,

CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

[43] D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y. Sung, B. Strope, R. Kurzweil, Universal Sentence Encoder, CoRR abs/1803.11175 (2018). URL: http://arxiv.org/abs/1803.11175. arXiv:1803.11175.

[44] C. J. Burges, From RankNet to LambdaRank to LambdaMART: An Overview, Learning 11 (2010) 81.

[45] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, CoRR abs/1907.11692 (2019). URL: http://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[46] M. Carnelos, L. Menotti, T. Porro, , G. Prando, Touché Task1: Argument Retrieval for Controversial Questions. Resolution provided by Team Goemon, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

[47] L. Gienapp, Quality-aware Argument Retrieval with Topical Clustering, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

[48] A. Mailach, D. Arnold, S. Eysoldt, S. Kleine, Exploring Document Expansion for Argument Retrieval, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

[49] M. Alecci, T. B. amd Luca Martinelli, , E. Ziroldo, Development of an IR System for Argument Search, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

[50] M. Lesk, Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone, in: V. DeBuys (Ed.), Proceedings of the 5th Annual International Conference on Systems Documentation, SIGDOC 1986, Toronto, Ontario, Canada, 1986, ACM, 1986, pp. 24–26. URL: https://doi.org/10.1145/318723.318728. doi:10.1145/318723.318728.

[51] R. Agarwal, A. Koniaev, R. Schaefer, Exploring Argument Retrieval for Controversial Questions Using Retrieve and Re-rank Pipelines, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

[52] N. Reimers, I. Gurevych, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, in: 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), Association for Computational Linguistics, 2019, pp. 3980–3990. URL: https://doi.org/10.18653/v1/D19-1410.

[53] E. D. Togni, A. Frasson, G. Zanatta, Exploring Approaches for Touché Task 1, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

[54] R. Krovetz, Viewing Morphology as an Inference Process, in: Proceedings of the 16th Annual International Conference on Research and Development in Information Retrieval (SIGIR 1993), Association for Computing Machinery, 1993, pp. 191–202. URL: https://doi.org/10.1145/160688.160718.

[55] 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.

[56] F. Berno, A. Cassetta, A. Codogno, E. Vicentini, , A. Piva, Shanks Touché Homework, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

[57] K. Ros, C. Edwards, H. Ji, C. Zhai, Argument Retrieval and Visualization, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

[58] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, L. Deng, MS MARCO: A Human Generated MAchine Reading COmprehension Dataset, in: Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches Co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), volume 1773 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2016. URL: http://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf.

[59] 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. doi:10.1145/383952.384019.

[60] T. Green, L. Moroldo, A. Valente, Exploring BERT Synonyms and Quality Prediction for Argument Retrieval, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

[61] M. Lippi, P. Torroni, MARGOT: A Web Server for Argumentation Mining, Expert Syst. Appl. 65 (2016) 292–303. URL: https://doi.org/10.1016/j.eswa.2016.08.050.

[62] S. Iyer, N. Dandekar, K. Csernai, First Quora Dataset Release: Question Pairs, 2017. Retrieved at https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs.

[63] J. Bevendorff, B. Stein, M. Hagen, M. Potthast, Elastic ChatNoir: Search Engine for the ClueWeb and the Common Crawl, in: Proceedings of the 40th European Conference on IR Research (ECIR 2018), volume 10772 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 820–824. URL: https://doi.org/10.1007/978-3-319-76941-7_83.

[64] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, in: Proceedings of the 1st International Conference on Learning Representations (ICLR 2013), 2013. URL: http://arxiv.org/abs/1301.3781.

[65] A. Trask, P. Michalak, J. Liu, sense2vec - A Fast and Accurate Method for Word Sense Disambiguation in Neural Word Embeddings, CoRR abs/1511.06388 (2015). URL: http://arxiv.org/abs/1511.06388. arXiv:1511.06388.

[66] T. Chen, C. Guestrin, XGBoost: A Scalable Tree Boosting System, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, 2016, pp. 785–794. URL: https://doi.org/10.1145/2939672.2939785.

[67] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T. Liu, LightGBM: A Highly Efficient Gradient Boosting Decision Tree, in: Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS 2017), 2017, pp. 3146–3154.

[68] J.-N. Weder, T. K. H. Luu, Argument Retrieval for Comparative Questions Based on

Independent Features, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

[69] V. Chekalina, A. Panchenko, Retrieving Comparative Arguments using Ensemble Methods and Neural Information Retrieval, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

[70] L. Breiman, Random Forests, Mach. Learn. 45 (2001) 5–32. URL: https://doi.org/10.1023/A:1010933404324.

[71] S. MacAvaney, OpenNIR: A Complete Neural Ad-Hoc Ranking Pipeline, in: Proceedings of the 13th ACM International Conference on Web Search and Data Mining (WSDM 2020), Association for Computing Machinery, 2020, pp. 845–848. URL: https://doi.org/10.1145/3336191.3371864.

[72] H. Hashemi, M. Aliannejadi, H. Zamani, W. B. Croft, ANTIQUE: A Non-factoid Question Answering Benchmark, in: Proceedings of the 42nd European Conference on IR Research (ECIR 2020), volume 12036 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 166–173. URL: https://doi.org/10.1007/978-3-030-45442-5_21.

[73] D. Helmrich, D. Streitmatter, F. Fuchs, M. Heykeroth, Touché Task 2: Comparative Argument Retrieval. A Document-based Search Engine for Answering Comparative Questions, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

[74] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language Models are Unsupervised Multitask Learners, OpenAI blog 1 (2019) 9.

[75] G. V. Cormack, M. D. Smucker, C. L. A. Clarke, Efficient and Effective Spam Filtering and Re-ranking for Large Web Datasets, Inf. Retr. 14 (2011) 441–465. URL: https://doi.org/10.1007/s10791-011-9162-z.

[76] A. Alhamzeh, M. Bouhaouel, E. Egyed-Zsigmond, J. Mitrović, DistilBERT-based Argumentation Retrieval for Answering Comparative Questions, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

[77] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter, CoRR abs/1910.01108 (2019). URL: http://arxiv.org/abs/1910.01108. arXiv:1910.01108.

[78] E. Shirshakova, A. Wattar, Thor at Touché 2021: Argument Retrieval for Comparative Questions, in: Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CLEF and CEUR-WS.org, 2021.

A. Full Evaluation Results of Touché 2021: Argument Retrieval

Table 5

Relevance results of all runs submitted to Task 1: Argument Retrieval for Controversial Questions. Reported are the mean nDCG@5 and the 95% confidence intervals. The two baseline rankings of the args.me search engine and DirichletLM are shown in bold.

Team	Run Tag	nDCG@5	CI95 Low	CI95 High
Elrond	ElrondKRun	0.720	0.651	0.785
Pippin Took	seupd2021-[…]-Dirichlet-mu-2000.0-topics-2021	0.705	0.634	0.772
Pippin Took	seupd2021-[…]-Dirichlet-mu-1500.0-topics-2021	0.702	0.626	0.767
Pippin Took	seupd2021-[…]-Dirichlet-mu-1800.0-topics-2021	0.701	0.632	0.770
Robin Hood	robinhood_combined	0.691	0.628	0.752
Pippin Took	seupd2021-[…]-Dirichlet-mu-2000.0-expanded-[…]	0.688	0.611	0.760
Pippin Took	seupd2021-[…]-Dirichlet-mu-1800.0-expanded-[…]	0.683	0.606	0.760
Asterix	run2021_Mixed_1.625_1.0_250	0.681	0.618	0.745
Dread Pirate Roberts	dreadpirateroberts_lambdamart_small_features	0.678	0.605	0.743
Asterix	run2021_Mixed_1.375_1.0_250	0.676	0.610	0.738
Elrond	ElrondOpenNlpRun	0.674	0.600	0.746
Asterix	run2021_Mixed_1.5_1.0_250	0.674	0.612	0.735
Elrond	ElrondSimpleRun	0.674	0.610	0.735
Robin Hood	robinhood_use	0.672	0.613	0.733
Skeletor	bm25-0.7semantic	0.667	0.598	0.733
Skeletor	manifold-c10	0.666	0.598	0.739
Skeletor	manifold	0.666	0.587	0.737
Asterix	run2021_Jolly_10.0_0.0_0.3_0.0__1.5_1.0_300	0.663	0.602	0.724
Luke Skywalker	luke-skywalker	0.662	0.598	0.732
Skeletor	bm25	0.661	0.581	0.732
Shanks	re-rank2	0.658	0.593	0.720
Heimdall	argrank_r1_c10.0_q5.0	0.648	0.580	0.715
Dread Pirate Roberts	dreadpirateroberts_lambdamart_medium_features	0.647	0.580	0.720
Robin Hood	robinhood_baseline	0.641	0.575	0.709
Heimdall	argrank_r1_c10.0_q10.0	0.639	0.569	0.710
Shanks	re-rank1	0.639	0.567	0.710
Shanks	LMDSimilarity	0.639	0.570	0.709
Athos	uh-t1-athos-lucenetfidf	0.637	0.568	0.705
Heimdall	argrank_r1_c5.0_q10.0	0.637	0.565	0.702
Goemon Ishikawa	goemon2021-dirichlet-lucenetoken-atirestop-nostem	0.635	0.561	0.704
Jean-Pierre Polnareff	seupd-jpp-dirichlet	0.633	0.570	0.699
Goemon Ishikawa	[…]-dirichlet-opennlptoken-terrierstop-nostem	0.630	0.558	0.698
Swordsman	**Dirichlet_multi_field**	**0.626**	**0.559**	**0.698**
Dread Pirate Roberts	dreadpirateroberts_dirichlet_filtered	0.626	0.554	0.691
Goemon Ishikawa	[…]-dirichlet-lucenetoken-terrierstop-[…]-queryexp	0.625	0.559	0.692
Yeagerists	run_4_chocolate-sweep-50	0.625	0.551	0.693
Yeagerists	run_2_lunar-sweep-201	0.624	0.547	0.698
Hua Mulan	args_naiveexpansion_0	0.620	0.556	0.688
Hua Mulan	args_gpt2expansion_0	0.620	0.550	0.686
Goemon Ishikawa	[…]-dirichlet-lucenetoken-lucenestop-nostem	0.620	0.552	0.689
Elrond	ElrondTaskBodyRun	0.614	0.544	0.680
Robin Hood	robinhood_rm3	0.611	0.532	0.688
Macbeth	macbethPretrainedBaseline	0.611	0.532	0.688
Yeagerists	run_3_lunar-sweep-58	0.610	0.541	0.681
Yeagerists	run_1_lucene_pure_rev	0.609	0.543	0.677
Macbeth	macbethBM25CrossEncoder	0.608	0.527	0.687
Macbeth	macbethBM25BiEncoderCrossEncoder	0.607	0.534	0.686
Swordsman	**args.me**	**0.607**	**0.528**	**0.676**
Goemon Ishikawa	[…]-dirichlet-lucenetoken-lucenestop-[…]-queryexp	0.607	0.539	0.679
Blade	bladeGroupBM25Method1	0.601	0.533	0.673
Shanks	multi-1	0.592	0.518	0.656
Shanks	multi-2	0.590	0.520	0.662
Blade	bladeGroupLMDirichlet	0.588	0.516	0.658
Dread Pirate Roberts	dreadpirateroberts_run_mlm	0.577	0.505	0.654
Skeletor	semantic	0.570	0.509	0.631
Asterix	run2021_Baseline_BM25	0.566	0.510	0.624
Dread Pirate Roberts	dreadpirateroberts_universal-sentence-encoder-qa	0.557	0.487	0.616
Deadpool	uh-t1-deadpool	0.557	0.476	0.631
Macbeth	macbethBM25AugmentedBiEncoderCrossEncoder	0.554	0.482	0.631
Yeagerists	run_5_good-sweep-85	0.536	0.456	0.612
Blade	bladeGroupBM25Method2	0.528	0.438	0.612
Batman	DE_RE_Analyzer_4r100	0.528	0.461	0.599
Little Foot	whoosh	0.521	0.442	0.596
Hua Mulan	args_t5expansion_0	0.518	0.448	0.581
Macbeth	macbethBiEncoderCrossEncoder	0.507	0.432	0.585
Gandalf	BM25F-gandalf	0.486	0.416	0.553
Palpatine	run	0.401	0.334	0.472
Batman	ER_v1	0.397	0.309	0.486
Heimdall	argrank_r0_c0.1_q5.0	0.004	0.000	0.013
Heimdall	argrank_r0_c0.01_q5.0	0.000	0.000	0.000
Batman	ER_Analyzer_5	0.000	0.000	0.000

Table 6

Quality results of all runs submitted to Task 1: Argument Retrieval for Controversial Questions. Reported are the mean nDCG@5 and the 95% confidence intervals. The two baseline rankings of the args.me search engine and DirichletLM are shown in bold.

Team	Run Tag	nDCG@5	CI95 Low	CI95 High
Heimdall	argrank_r1_c10.0_q10.0	0.841	0.802	0.876
Heimdall	argrank_r1_c5.0_q10.0	0.839	0.803	0.875
Heimdall	argrank_r1_c10.0_q5.0	0.833	0.797	0.869
Skeletor	manifold	0.827	0.783	0.868
Skeletor	bm25	0.822	0.784	0.861
Skeletor	manifold-c10	0.818	0.778	0.856
Asterix	run2021_Jolly_10.0_0.0_0.3_0.0__1.5_1.0_300	0.818	0.783	0.853
Elrond	ElrondOpenNlpRun	0.817	0.777	0.856
Skeletor	bm25-0.7semantic	0.815	0.774	0.852
Asterix	run2021_Mixed_1.375_1.0_250	0.814	0.774	0.853
Pippin Took	seupd2021-[…]-Dirichlet-mu-1800.0-expanded-[…]	0.814	0.773	0.852
Pippin Took	seupd2021-[…]-Dirichlet-mu-2000.0-expanded-[…]	0.814	0.774	0.850
Goemon Ishikawa	goemon2021-dirichlet-lucenetoken-atirestop-nostem	0.812	0.767	0.854
Hua Mulan	args_gpt2expansion_0	0.811	0.773	0.849
Dread Pirate Roberts	dreadpirateroberts_lambdamart_medium_features	0.810	0.769	0.849
Yeagerists	run_4_chocolate-sweep-50	0.810	0.771	0.848
Elrond	ElrondKRun	0.809	0.765	0.853
Yeagerists	run_2_lunar-sweep-201	0.809	0.773	0.846
Robin Hood	robinhood_baseline	0.809	0.770	0.844
Luke Skywalker	luke-skywalker	0.808	0.767	0.850
Asterix	run2021_Mixed_1.5_1.0_250	0.807	0.764	0.848
Yeagerists	run_5_good-sweep-85	0.807	0.768	0.844
Goemon Ishikawa	[…]-dirichlet-lucenetoken-terrierstop-[…]-queryexp	0.806	0.764	0.845
Dread Pirate Roberts	dreadpirateroberts_lambdamart_small_features	0.804	0.765	0.844
Robin Hood	robinhood_rm3	0.804	0.755	0.850
Macbeth	macbethBM25CrossEncoder	0.803	0.762	0.840
Athos	uh-t1-athos-lucenetfidf	0.802	0.758	0.844
Jean-Pierre Polnareff	seupd-jpp-dirichlet	0.802	0.763	0.838
Asterix	run2021_Mixed_1.625_1.0_250	0.802	0.758	0.843
Pippin Took	seupd2021-[…]-Dirichlet-mu-1800.0-topics-2021	0.799	0.760	0.838
Yeagerists	run_3_lunar-sweep-58	0.799	0.760	0.838
Yeagerists	run_1_lucene_pure_rev	0.798	0.755	0.837
Pippin Took	seupd2021-[…]-Dirichlet-mu-2000.0-topics-2021	0.798	0.758	0.839
Goemon Ishikawa	[…]-dirichlet-opennlptoken-terrierstop-nostem	0.797	0.757	0.836
Pippin Took	seupd2021-[…]-Dirichlet-mu-1500.0-topics-2021	0.797	0.755	0.834
Goemon Ishikawa	[…]-dirichlet-lucenetoken-lucenestop-nostem	0.796	0.756	0.837
Swordsman	**Dirichlet_multi_field**	**0.796**	**0.759**	**0.837**
Dread Pirate Roberts	dreadpirateroberts_dirichlet_filtered	0.796	0.757	0.839
Goemon Ishikawa	[…]-dirichlet-lucenetoken-lucenestop-[…]-queryexp	0.796	0.756	0.836
Shanks	re-rank1	0.795	0.754	0.836
Shanks	LMDSimilarity	0.795	0.757	0.835
Shanks	re-rank2	0.790	0.750	0.826
Hua Mulan	args_naiveexpansion_0	0.789	0.747	0.830
Elrond	ElrondTaskBodyRun	0.788	0.742	0.830
Macbeth	macbethPretrainedBaseline	0.783	0.738	0.824
Macbeth	macbethBM25BiEncoderCrossEncoder	0.783	0.743	0.828
Dread Pirate Roberts	dreadpirateroberts_run_mlm	0.779	0.737	0.820
Heimdall	argrank_r0_c0.1_q5.0	0.767	0.725	0.811
Blade	bladeGroupLMDirichlet	0.763	0.706	0.815
Robin Hood	robinhood_combined	0.756	0.708	0.806
Macbeth	macbethBM25AugmentedBiEncoderCrossEncoder	0.752	0.704	0.801
Blade	bladeGroupBM25Method1	0.751	0.705	0.799
Macbeth	macbethBiEncoderCrossEncoder	0.750	0.701	0.802
Heimdall	argrank_r0_c0.01_q5.0	0.749	0.707	0.793
Elrond	ElrondSimpleRun	0.740	0.693	0.785
Robin Hood	robinhood_use	0.732	0.680	0.782
Little Foot	whoosh	0.718	0.661	0.766
Swordsman	**args.me**	**0.717**	**0.663**	**0.773**
Blade	bladeGroupBM25Method2	0.705	0.639	0.766
Batman	DE_RE_Analyzer_4r100	0.695	0.638	0.751
Shanks	multi-2	0.684	0.627	0.739
Deadpool	uh-t1-deadpool	0.679	0.618	0.738
Shanks	multi-1	0.674	0.616	0.728
Skeletor	semantic	0.671	0.602	0.737
Asterix	run2021_Baseline_BM25	0.671	0.619	0.721
Batman	ER_Analyzer_5	0.671	0.598	0.741
Batman	ER_v1	0.662	0.589	0.721
Hua Mulan	args_t5expansion_0	0.654	0.584	0.727
Dread Pirate Roberts	dreadpirateroberts_universal-sentence-encoder-qa	0.624	0.558	0.681
Gandalf	BM25F-gandalf	0.603	0.532	0.672
Palpatine	run	0.562	0.497	0.633

Table 7

Relevance results of all runs submitted to Task 2: Argument Retrieval for Comparative Questions. Reported are the mean nDCG@5 and the 95% confidence intervals; ChatNoir baseline in bold.

Team	Run Tag	nDCG@5	CI95 Low	CI95 High
Katana	py_terrier_xgb	0.489	0.421	0.557
Thor	uh-t2-thor	0.478	0.400	0.563
Rayla	DistilBERT_argumentation_advanced_ranking_run_1	0.473	0.409	0.540
Rayla	DistilBERT_argumentation_advanced_ranking_run_3	0.471	0.399	0.538
Jack Sparrow	Jack Sparrow__bert	0.467	0.396	0.533
Rayla	DistilBERT_argumentation_bm25	0.466	0.392	0.541
Katana	lgbm_ranker	0.460	0.395	0.531
Rayla	DistilBERT_argumentation_advanced_ranking_run_2	0.458	0.395	0.525
Mercutio	ul-t2-mercutio-run_2	0.441	0.374	0.503
Jack Sparrow	Jack Sparrow_	0.422	0.357	0.489
Puss in Boots	**ChatNoir**	**0.422**	**0.354**	**0.490**
Katana	rand_forest	0.393	0.328	0.461
Katana	run_tf.txt	0.385	0.320	0.456
Katana	run.txt	0.377	0.311	0.445
Mercutio	ul-t2-mercutio-run_1	0.372	0.306	0.438
Jack Sparrow	Jack Sparrow__argumentative_bert	0.341	0.293	0.391
Jack Sparrow	Jack Sparrow__argumentative	0.340	0.277	0.408
Mercutio	ul-t2-mercutio-run_3	0.320	0.258	0.386
Prince Caspian	prince-caspian	0.244	0.174	0.321
Katana	bert_test	0.091	0.057	0.127

Table 8

Quality results of all runs submitted to Task 2: Argument Retrieval for Comparative Questions. Reported are the mean nDCG@5 and the 95% confidence intervals; ChatNoir baseline in bold.

Team	Run Tag	nDCG@5	CI95 Low	CI95 High
Rayla	DistilBERT_argumentation_bm25	0.688	0.614	0.758
Katana	lgbm_ranker	0.684	0.624	0.749
Thor	uh-t2-thor	0.680	0.606	0.760
Katana	py_terrier_xgb	0.675	0.605	0.740
Rayla	DistilBERT_argumentation_advanced_ranking_run_1	0.670	0.592	0.743
Jack Sparrow	Jack Sparrow__bert	0.664	0.596	0.735
Jack Sparrow	Jack Sparrow_	0.652	0.582	0.718
Mercutio	ul-t2-mercutio-run_2	0.651	0.577	0.728
Puss in Boots	**ChatNoir**	**0.636**	**0.559**	**0.713**
Katana	run_tf.txt	0.630	0.560	0.702
Rayla	DistilBERT_argumentation_advanced_ranking_run_2	0.630	0.542	0.709
Katana	rand_forest	0.628	0.558	0.691
Rayla	DistilBERT_argumentation_advanced_ranking_run_3	0.625	0.548	0.696
Jack Sparrow	Jack Sparrow__argumentative_bert	0.620	0.568	0.667
Mercutio	ul-t2-mercutio-run_1	0.610	0.537	0.679
Katana	run.txt	0.608	0.537	0.673
Jack Sparrow	Jack Sparrow__argumentative	0.606	0.542	0.668
Prince Caspian	prince-caspian	0.548	0.457	0.630
Mercutio	ul-t2-mercutio-run_3	0.530	0.454	0.600
Katana	bert_test	0.466	0.388	0.542