DistilBERT-based Argumentation Retrieval for Answering Comparative Questions

Notebook for the Touché Lab on Argument Retrieval at CLEF 2021

Alaa Alhamzeh[1,2], Mohamed Bouhaouel[1], Elősd Egyed-Zsigmond[2] and Jelena Mitrović[1]

[1] *Universität Passau, Innstraße 41, 94032 Passau, Germany*

[2] *INSA de Lyon, 20 Avenue Albert Einstein, 69100 Villeurbanne, France*

Abstract

In the current world, individuals are faced with decision making problems and opinion formation processes on a daily basis. For example, debating or choosing between two similar products. However, answering a comparative question by retrieving documents based only on traditional measures (such as TF-IDF and BM25) does not always satisfy the need. Thus, introducing the argumentation aspect in the information retrieval procedure recently gained significant attention. In this paper, we present our participation at the CLEF 2021 Touché Lab for the second shared task, which tackles answering comparative questions based on arguments. Therefore, we propose a novel multi-layer architecture where the argument extraction task is considered as the main engine. Our approach therefore is a pipeline of query expansion, argument identification based on DistilBert model, and sorting the documents by a combination of different ranking criteria.

Keywords

Comparative Question Answering, Argumentation Mining, Transfer Learning, Argument Identification, Computational Linguistic

1. Introduction

Argumentation is a fundamental aspect of human communication and decision making. It can be defined as the logical reasoning humans use to derive a conclusion or justify their perspectives on a specific topic.

An argument consists of two elementary components: one or more premises and one claim (conclusion) [1]. Argument mining is the automatic identification or extraction of arguments from unlabeled input text such as news articles, scientific papers, legal documents, reviews, and debates. Hence, argument mining has been investigated in various applications such as assisted writing and legal counselling. Furthermore, the arguments can serve as the keystone of an intelligent web search engine for question answering. In this context, the Webis Group [1] organized an argumentation retrieval event "Touché Lab at CLEF 2021" [2] that consists of two

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

alaa.alhamzeh@insa-lyon.fr (A. Alhamzeh); mohamed.bouhaouel@uni-passau.de (M. Bouhaouel); Elod.Egyed-zsigmond@insa-lyon.fr (E. Egyed-Zsigmond); Jelena.Mitrovic@uni-passau.de (J. Mitrović)

CEUR Workshop Proceedings (CEUR-WS.org)

[1] https://webis.de/

[2] https://webis.de/events/touche-21/index.html

independent shared tasks: 1) Argument Retrieval for Controversial Questions and 2) Argument Retrieval for Comparative Questions. We present in this paper our adopted approach for participation as **"Rayla Team"** in the latter shared task. The main objective of this task is to support users facing some choice problem in their daily life. Given a comparative question (for instance, "Which is better, a Mac or a PC?"), the goal is to retrieve documents from the ClueWeb12 corpus [3], and rank them based on different criteria, mainly, the arguments they provide.

In order to have more granularity control, our proposed architecture incorporates several units, each one performs a specific sub-task, namely: query expansion, argument extraction, scoring, and sorting.

The Touché organizers gave the option to use their own tool, TARGER[2], for the argument retrieval sub-task by providing a simple REST API to interact with. However, since this is the main block in our proposed pipeline, we decided to develop our own module based on the latest developments in the field of deep learning. Therefore, we implemented a new transfer learning model for Argument Identification based on DistilBert [3], a distilled version of BERT (Bidirectional Encoder Representations from Transformers) [4]: smaller, faster, and lighter, with 97% of its language understanding capabilities [3].

The remaining of the paper is organized as follows: In Section 2, we go through a conceptual background of argument mining and the recent usage of transfer learning models towards it . In Section 3, we present our retrieving system architecture as well as the function of each unit. We discuss the proposed approach in Section 5. Finally, we summarize the main achievements in Section 6.

2. Related work

2.1. Argument Retrieval

Argumentative text identification is generally the first step of a complete argument mining system, then comes the argument components detection, and finally, the argument structure identification that considers the support and attack relations. State-of-the-art literature reports mainly classical machine learning models and very few attempts to use deep learning models (e.g. [5, 6, 7]).

In a classical machine learning model, the training and testing data are used for the same task and follow the same distribution. However, transfer learning aims to transfer previous learned knowledge from one source task (or domain) to a different target one, considering that source and target tasks and domains may be the same or may be different but related[8].

This can be useful in the argument mining domain from different perspectives. First of all, common knowledge about the language is obviously appreciated. Second, transfer learning can solve or at least help to solve one of the biggest challenges in the argument mining field, the lack of labeled datasets. Third, even available datasets are often of small size and very domain and task dependent. They may follow different annotations, argument schemes, and various feature spaces. This means that in each potential application of argument mining, we need

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

argument experts to label a significant amount of data for the task at hand, which is definitely an expensive work in terms of time and human-effort. Hence, transfer learning will fine-tune pre-trained knowledge on a big dataset to serve another problem, and that's why we built on it for the argument identification stage.

Looking at the literature, we found only two studies addressing transfer learning models for argumentation tasks published in 2020. The first one is towards discriminating evidence related to Argumentation Schemes [9] where the authors train classifiers on the sentence embeddings extracted from different pre-trained transformers. The second one is by Wambsganss et al. [5] where the authors proposed an approach for argument identification using BERT[4]. Our transfer learning model is based on a distilled version of BERT, proposed by [3], which retains 97% of the language understanding capabilities of the base BERT model, with a 40% less in size, and being 60% faster.

2.2. Comparative Question Answering

Question answering is a sophisticated form of information retrieval that started to develop as a field decades ago. However, with the growth of world wide web data, as well as of private databases, the need for more precise answers, well-expressed and shortly formulated is growing, too. Hence, several studies are devoted to the representation of natural language stated in the query and in the documents.

Extracting the arguments stated in the document is one way to clearly capture the grounded statements (premises) and the final conclusion (claim) presented in the text. Therefore, many recent works focus on the arguments as a potential tool for improving comparative question answering [10, 11]. Improving a retrieval model for argument-based comparison systems is the target of the Touché's Task 2, which has been addressed in 2020 [12] as well.

The best result of last year was introduced by the team "Bilbo Baggins" of Abye et al. [13]. They used a two-step approach: 1) query expansion by antonyms and synonyms, and 2) document ranking based on three measures: relevance, credibility, and support features. An overall score is deduced by summing up those three scores after weights multiplication. Eventually this approach can be improved by keeping the same pipeline and enhance the inner sub-tasks. For instance, instead of using a static classifier 'XGBoost' for the comparative sentence classification, a BERT-based model can ameliorate the robustness of the classifier and extend its operating range. In [14], Johannes Huck participated with a simple approach, the original topic is used as a single query to be sent to ChatNoir, and retrieve the top 20 documents. After documents retrieval, the author uses the TARGER API [2] in order to extract arguments from each document, claims and premises are treated as arguments without any distinction. For the re-ranking, the BM25 algorithm has been used to determine the relevance of the extracted arguments with respect to the original query [14]. Despite the simplicity of this implementation and the lack of introducing several measures in the ranking process, this approach could be integrated as one component in a bigger architecture in order to add an extra measure that may improve the stability of the overall system.

3. Approach and Implementation

In this section, we present our proposed approach and adopted methods to build a search engine for answering comparative questions based on argumentation identification. The overall architecture of our approach is presented in Figure 1. It composes of a sequence of seven stages. We go through them individually in the upcoming sections. We used the same architecture to submit four runs with different configurations via TIRA platform [15].

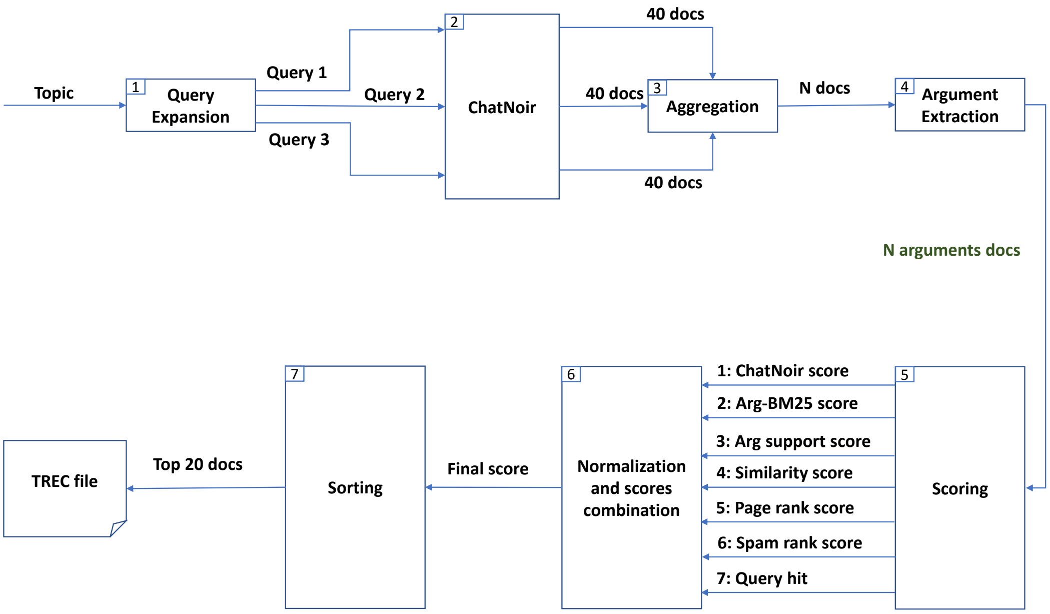

Figure 1: Global architecture of the submitted approach

3.1. Query Expansion

Query Expansion (QE) is the process of reformulating a given query to improve retrieval performance and increase the recall of the candidate retrieval (i.e. to match additional documents). Our query expansion module involves a variety of techniques in order to generate three different queries to be passed to the next step, as the following:

- *Query 1:* is the original query itself.
- *Query 2:* is generated from the original query by: (1) removing English stop words, punctuation marks, and comparison adjectives or also called comparison operators. (2) Stemming of the remaining words to their base forms, and aggregating them together with conjunctive AND operator.

- *Query 3:* will be generated from the original query only if the latter contains a comparison operator, as follows:
  - Search for synonyms and/or antonyms of the comparison operator of the query to get what is called the *context of the comparison operator*, whose size is 5 synonyms/antonyms in our case.
  - Remove English stop words and punctuation.
  - Eliminate the comparison operator from the original query and transform the remaining words/terms to their base form.
  - We create 5 queries out of the original query (after the pre-specified changes: removing English stop words, punctuation and comparison operators), and each time adding one of the 5 synonyms/antonyms from the *context of the comparison operator*. Those 5 output queries are sent to ChatNoir as one disjunctive OR-query: Query 3.

Comparison adjectives identification and words stemming are done automatically using using spaCy[4]. Synonyms and antonyms are hard-coded in the software due to the limited number of comparative operators in the topics of this year and the last year (10 adjectives in total). In Table 1 we show an example of the query expansion output for the Topic 61 from Touché 2021 topics.

Table 1
Example of query expansion

Query id	Generated query
Query 1	Who is stronger, Hulk or Superman?
Query 2	Hulk *AND* Superman
Query 3	Hulk *AND* Superman *AND* strong *OR* Hulk *AND* Superman *AND* capable *OR* Hulk *AND* Superman *AND* powerful *OR* Hulk AND Superman *AND* able *OR* Hulk *AND* Superman *AND* weak

3.2. Retrieving the Documents by ChatNoir API

The ClueWeb12 document dataset for this task is easily accessible through the ChatNoir API [5] [16, 17] that is based on the BM25F ranking algorithm. The BM25, also known as *Best Matching 25* is a ranking method used by several search engines to determine the relevance of documents with regards to a query. It is based on the probabilistic retrieval framework and it is considered as the state-of-the-art among TF-IDF-like algorithms in information retrieval. BM25F is just a newer version of BM25 that takes the structure of documents into consideration.

[4] https://spacy.io/

[5] https://www.chatnoir.eu/doc/api/

The API takes a query as input, in our case we retrieve 40 documents for each of the 3 queries. The interaction with this API can be done using either GET or POST requests, several parameters could be included in the request to specify the corpus to retrieve from, and the type of returned information. It supports also a set of standard operators from web search services, and queries can be concatenated using *Boolean operators, -, "…", site:… etc.*. In our approach, we used only POST requests, *AND* and *OR* operators to concatenate strings in Query 2 and Query 3.

3.3. Documents Aggregation

As Query 3 may be empty, we have a minimum of 80 and a maximum of 120 retrieved documents. Every document has a unique id (uuid), which we use to remove the redundant documents returned by more than one query. As a result we get N documents. One important remark at this phase, is that when removing a redundant document, we preserve its importance by including its retrieved scores (from ChatNoir) to the scores of the remaining unique document (having a unique uuid). This is done by summing up individual scores to obtain a single score for each set of redundant documents.

Initially, the ChatNoir API does not respond with the full content of the documents. Instead, it returns some metadata, including the unique uuid. To get a document full content, a new query is sent to ChatNoir to request the full HTML source page. Later on, we need to reveal only the main textual content from the HTML document. We achieve that by removing tags, advertisements and other cleaning processes. For that end, we used two HTML cleaning libraries: *Boilerpy3* [6] and *Trafilatura* [7].

Both libraries are powerful and able to perform the main text extraction most of the time. However, the output of the two libraries is not always the same. In some cases, *Trafilatura* performs better than *Boilerpy3* and extracts more text content, while the latter outperforms in other cases/documents. For this reason, we are using both libraries to extract the HTML content, the results are split into sentences and we take the union of those sentences in order to cover the entire main content of the document. The final list of sentences is sent to the *Argument Extraction* module, where each document is presented as a set of sentences.

3.4. Argument Extraction

In our particular task, we seek to detect the comparative sentences in the document, therefore, argument identification can be sufficient. Hence, we take the sentences from the document aggregation step and apply binary classification using the model presented in Figure 2 to label every sentence as an argument or Non-argument. Therefore, we searched for datasets that contain both argument and non-argument labels. Student Essays [18] and Web discourse [19] are public, very common used corpora which well satisfy this purpose.

The **Student Essays corpus** contains 402 Essays about 8 controversial topics. The annotation covers the following argument components: 'major claim', 'claim' and 'premise'. In addition, it

[6] https://github.com/jmriebold/BoilerPy3

[7] https://github.com/adbar/trafilatura

shows the support/attack relations between them. Thus, it could be used in several argument mining tasks.

The **User-generated Web Discourse corpus** is a smaller dataset that contains 340 documents about 6 controversial topics in education such as homeschooling. The document may refer to an article, blog post, comment, or forum posts. This dataset is considered noisy, unrestricted and less formalized. The annotation has been done by [19] according to Toulmin's model [20]. Thus, it covers the following argument components: 'claim', 'premise', 'backing' and 'rebuttal'.

In order to deduct binary-labeled unified data for both corpora, we label any argument component (premise or claim) as an 'argument', and the rest of the text sentences as 'non-argument'.

The choice of BERT-based model is justified by the fact that among different existent transformers, BERT [4] has recently gained a lot of attention. It seems to achieve the state of the art results in several NLP tasks [5, 6, 7, 21].

For our particular task, we performed different experiments using several BERT-based models (BERT base, RoBERTa-base[22], DistilRoBERTa , DistilBERT)[8] and we achieved very similar results. Hence, we decided to use the DistilBERT model given that it is 40% less than BERT in size with a relevant in-line performance and a faster training/ testing time [3].

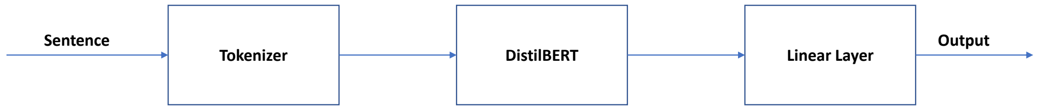

Figure 2: Transfer Learning Model Architecture

Figure 2 shows our transfer learning model architecture to perform the argument identification task using DistilBERT. The first block is the Tokenizer that takes care of all the BERT input requirements, by doing the following: (1) It transforms the words of a sentence into an array of DistilBERT tokens. (2) It adds the special starting token ([CLS] token). (3) It adds the necessary padding to preserve the unique size for all sentences (we set 128 as a maximum length). The second block is the DistilBERT fine-tuned model that outputs mainly a vector of length of 768 (default length). In order to adapt the output of this pre-trained model to our specific task, we add a third block, which is a linear layer applied on top of the DistilBERT transformer, and outputs a vector of size 2. In this vector, the index of the maximum value reflects the predicted class id (Or simply it is the $argmax$ of the output vector). We trained the model for 3 epochs, using AdamW [23] as an optimizer and Cross Entropy for the loss calculation.

3.5. Scoring

This step is essential for any search engine system because users usually check out the first result links. Therefore, our objective now is to estimate the best matching between the query

[8] We used Transformers from huggingface.com for our experiments.

Table 2
DistilBERT results on different datasets

Dataset	Accuracy	Precision	Recall	F1-score
Student Essays	0.8727	0.8016	0.7477	0.7697
Web Discourse	0.7799	0.7718	0.6484	0.6655
Merged Corpora	0.8587	0.7887	0.7529	0.7683

and the candidate answers, in order to sort them at the final stage. For that end, we investigate different scores based on different aspects. First of all, the document relevance which can be checked by ChatNoir BM25 score and Query hit count.

However, even if a document content is relevant to the query, it may be faked or biased, we inspect the credibility of the document itself by considering: Page Rank score as well as Spam rank score.

Moreover, as we built our retrieval system based on arguments, we takes into consideration the argument quality level by three different scores: argument support, query-argument similarity, and argument BM25 score.

The complete details of our ranking scores are:

1. ChatNoir score: Returned form ChatNoir API indicating BM25 measure.
2. Query hit count: Indicates how many times the document is retrieved by the three queries[1,3].
3. Page Rank score: Returned form ChatNoir API measuring the importance of the source website pages.
4. Spam rank score: Returned form ChatNoir API indicating the probability of the website to be a spam.
5. Argument support score: Represents the ratio of argument sentences among all existent sentences in the document.
6. Similarity score: Evaluates the similarity of two sentences based on the context and English language understanding using the *SentenceTransformer* [9] library [24]. We calculate the similarity between the original query and every argumentative sentence in the document, and consider the average as the score.
7. Arg-BM25 score: Calculated on argumentative sentences of each document with respect to the original query. This is done through re-indexing the retrieved documents by creating new ones that contain only argumentative sentences. Then the arg-BM25 score of each document is calculated by querying the new argumentative documents with the original topic.

3.6. Normalization and Scores Combination

Furthermore, for the final score, we normalize all previously calculated scores, so that all values are between 0 and 1. These scores are summed up using particular weights. The setting up of the scoring weights is fixed manually based on the announced relevance judgments of last year

[9] https://github.com/UKPLab/sentence-transformers

Table 3
Configurations of each run

	Score Weights							
Run Tag	(1)	(2)	(3)	(4)	(5)	(6)	(7)	Docs
DistilBERT_argumentation_bm25	0	1	0	0	0	0	0	30
DistilBERT_argumentation_advanced_ranking_r1	15	25	25	15	20	0	0	20
DistilBERT_argumentation_advanced_ranking_r2	10	10	50	20	10	5	5	40
DistilBERT_argumentation_advanced_ranking_r3	10	15	10	50	10	0	0	40

Touche task 2. Thereby, we have done several experiments on the topics from last year, while changing each time the values of weights. Then, we took the best values of weights and applied them in different runs for this year submission.

3.7. Sorting

It this step, the documents are sorted based on the final score to get the top 20 documents that are highly relevant to answer the comparative topics. The final output is inserted into a text file while respecting the standard TREC format proposed by the Touché organization, so that each line should contain the following fields:
qid Q0 doc rank score tag

- qid: The topic ID
- Q0: Reserved (Should always be Q0)
- doc: The document ID, which is represented by the *trec_id* retrieved from ChatNoir
- rank: The final rank of the document
- score: The final score of the document
- tag: Short text to identify the group and the used method

4. Evaluation

The shown architecture in Figure 1 presents our base approach, from which we derive four submissions to the task-2 of Touché 2021 by modifying score weights and the number of retrieved and submitted documents. In Table 3, we present the used score weights for each run in addition to the number of submitted documents per topic, scores id are defined in Section 3.5.

The results of each submission are conducted by the Touché committee through the manual labeling of the documents with the help of human assessors. For each run, two evaluation scores are calculated that defines the relevance and the rhetorical quality of the submitted documents. For the quality evaluation, our team Rayla comes out on top of the ranking list by achieving a score of *nDCG@5*= 0.675. While we are ranked third with respect to the relevance evaluation with an *nDCG@5* score of 0.427. Table 4 presents in more details the achievements of each run in more details.

Table 4
Achievements of each run

	Relevance		Quality	
Run Tag	nDCG@5	Rank ./20	nDCG@5	Rank ./20
DistilBERT_argumentation_bm25	0.466	6	0.688	1
DistilBERT_argumentation_advanced_ranking_r1	0.473	3	0.670	5
DistilBERT_argumentation_advanced_ranking_r2	0.458	8	0.630	11
DistilBERT_argumentation_advanced_ranking_r3	0.471	4	0.625	13

5. Discussion

The proposed approach presents an advanced web search engine to answer comparative questions, each component in the architecture presented in Figure 1 plays an important role to achieve the best retrieval results. For instance, the query expansion component tries to make a first selection and build a set of topic-related documents. The three different queries generated from the original topic increase the coverage of the related documents and this works very well with the ChatNoir API since it is a basic BM25 retrieval system.

Table 2 presents the results of our DistilBERT model to accomplish the binary classification task (argument / non-argument) at the sentence level on different corpora. It shows that our DistilBERT model results are in-line to recent work of Wambsganss. et al. [5], it achieves very good results in argument classification with an accuracy of 0.8587 with respect to the merged corpora. Furthermore, the model is built on top of a text processing transformer, DistilBERT, and it is powered by English knowledge understanding, which makes the trained argument identification model scale on a wide variety of English topics and texts.

In order to re-rank the initially retrieved documents, as mentioned in Section 3.5, we included several measures in the final score calculation. Those measures define the importance of the document in terms of three criteria: (1) document relevance, (2) whether sufficient argumentative support is provided, and (3) trustworthiness and credibility. For the document relevance criteria, the *ChatNoir score* and the *query-hit* reflect well whether a document is relevant enough to the topic, while the *spam rank score* and the *page rank score* reveal the trustworthiness and credibility of the web page. For the argument support, three different scores are being considered, *argument support score*, *argument similarity score* and *argument BM25 score.*

6. Conclusion

In this paper, we presented our participation at the shared task of argument retrieval for answering comparative questions. Our approach consists of a pipeline of several components for query expansion, retrieving the documents, extracting the arguments, and ranking them respectively. A main contribution is the transfer learning model we developed based on the DistilBert transformer, and using two different datasets for adapting it to the argument identification task. In addition, we consider different criteria to re-rank the output documents with respect to their relevance, the argument they contain, and their trustworthiness and credibility. In future work, we plan to use more advanced techniques for better query enrichment, in addition to using a machine learning model that learns the best weights of the scores mentioned in Section 3.5.

References

[1] T. Govier, A practical study of argument, 7th ed. ed., Cengage Learning, Belmont, CA, 2010.

[2] A. Chernodub, O. Oliynyk, P. Heidenreich, A. Bondarenko, M. Hagen, C. Biemann, A. Panchenko, Targer: Neural argument mining at your fingertips, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2019, pp. 195–200.

[3] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108 (2019).

[4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[5] T. Wambsganss, N. Molyndris, M. Söllner, Unlocking transfer learning in argumentation mining: A domain-independent modelling approach, in: 15th International Conference on Wirtschaftsinformatik, 2020.

[6] N. Reimers, B. Schiller, T. Beck, J. Daxenberger, C. Stab, I. Gurevych, Classification and clustering of arguments with contextualized word embeddings, arXiv preprint arXiv:1906.09821 (2019).

[7] T. Niven, H.-Y. Kao, Probing neural network comprehension of natural language arguments, arXiv preprint arXiv:1907.07355 (2019).

[8] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (2009) 1345–1359.

[9] D. Liga, M. Palmirani, Transfer learning with sentence embeddings for argumentative evidence classication, in: F. Grasso, N. L. Green, J. Schneider, S. Wells (Eds.), Proceedings of the 20th Workshop on Computational Models of Natural Argument co-located with the 8th International Conference on Computational Models of Argument (COMMA 2020), Perugia, Italy (and online), September 8th, 2020, volume 2669 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 11–20. URL: http://ceur-ws.org/Vol-2669/paper2.pdf.

[10] A. Bondarenko, A. Panchenko, M. Beloucif, C. Biemann, M. Hagen, Answering comparative questions with arguments, Datenbank-Spektrum 20 (2020) 155–160. doi:10.1007/s13222-020-00346-8.

[11] A. Panchenko, A. Bondarenko, M. Franzek, M. Hagen, C. Biemann, Categorizing comparative sentences, arXiv preprint arXiv:1809.06152 (2018).

[12] A. Bondarenko, L. Gienapp, M. Fröbe, M. Beloucif, Y. Ajjour, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, M. Potthast, M. Hagen, Overview of Touché 2021: Argument Retrieval, in: D. Hiemstra, M.-F. Moens, J. Mothe, R. Perego, M. Potthast, F. Sebastiani (Eds.), Advances in Information Retrieval. 43rd European Conference on IR Research (ECIR 2021), volume 12036 of *Lecture Notes in Computer Science*, Springer, Berlin Heidelberg New York, 2021, pp. 574–582. URL: https://link.springer.com/chapter/10.1007/978-3-030-72240-1_67. doi:10.1007/978-3-030-72240-1_67.

[13] T. Abye, T. Sager, A. J. Triebel, An open-domain web search engine for answering com-

parative 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.

[14] 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.

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

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

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

[18] C. Stab, I. Gurevych, Annotating argument components and relations in persuasive essays, in: Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers, 2014, pp. 1501–1510.

[19] I. Habernal, I. Gurevych, Argumentation mining in user-generated web discourse, Computational Linguistics 43 (2017) 125–179.

[20] S. E. Toulmin, The Uses of Argument, 2 ed., Cambridge University Press, 2003. doi:10.1017/CBO9780511840005.

[21] T. Caselli, V. Basile, J. Mitrović, I. Kartoziya, M. Granitzer, I Feel Offended, Don't Be Abusive! Implicit/Explicit Messages in Offensive and Abusive Language, in: Proceedings of LREC, 2020.

[22] 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, arXiv preprint arXiv:1907.11692 (2019).

[23] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net, 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.

[24] 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, Association for Computational Linguistics, 2019. URL: https://arxiv.org/abs/1908.10084.