Ranking Arguments by Combining Claim Similarity and Argument Quality Dimensions

Notebook for Touché: Argument Retrieval at CLEF 2020

Lorik Dumani and Ralf Schenkel

Trier University, Germany
{dumani,schenkel}@uni-trier.de

Abstract In this paper we describe our submissions to the CLEF lab Touché, which addresses argument retrieval from a focused debate collection. Our approach consists of a two-step retrieval. Step one finds the most similar claims to a query. Step two ranks the directly tied premises by the count of their convincingness compared to other relevant premises, for which we aggregate the sum of three main argument quality dimensions. The final ranking consists of the product of the two components which are expressed as probabilities.

1 Introduction

Argumentation is required not only in political debates, where people try to convince others of certain standpoints, e.g., political views. They are also essential for personal decision making, e.g., which smartphone to buy. Since the emergence of well-equipped computers and the increasingly sophisticated NLP methods, computational argumentation has become a very popular field of research and seeks to help people to find good and strong arguments for their needs. In line with existing work, an *argument* is defined as a *claim* supported or attacked by at least one *premise* [13]. The claim is usually a controversial standpoint that should not be believed by a reader without further evidence (in form of premises).

Touché [5,4] is the first lab on Argument Retrieval.[1] It follows the classical TREC-style[2] evaluation methodology and features two subtasks:

1. Argument retrieval from a focused debate collection to support argumentative conversations by providing justifications for the claims.
2. Argument retrieval from a generic Web crawl to answer comparative questions with argumentative results and to support decision making.

We participated in Task 1 and in this paper we provide a description of the implementation of our approach. Our submissions to the task were done under the team name Don Quixote.

[1] https://events.webis.de/touche-20/.

[2] https://trec.nist.gov/tracks.html.

Task 1 aims at supporting users directly and finding arguments, e.g., to strengthen their stance or to form opinions about certain issues by "strong" arguments. Thus, besides general relevance of the argument to the topic, *argument quality dimensions* (see the work of Wachsmuth et al. [15] for a survey of work on argument quality) will be also evaluated. From the top-k results of all submissions, pools of answers will be formed that will be assessed by crowd-sourcing; evaluation of the submissions will then be done with nDCG [10]. The retrieved arguments will be evaluated by the following qualities:

(1) whether an argumentative text is *logically cogent*,
(2) whether it is *rhetorically well-written*, and
(3) whether it contributes to the users' stance-building process (called *utility*).

The Web provides innumerable documents that contain arguments. Especially in online portals controversial topics are discussed. Since basically anyone can participate in the discussion, we can assume that practically all relevant aspects are addressed there. Furthermore, as most of the participants are not experts, the arguments might be written in an understandable language. Hence, the official data basis is the dataset from Ajjour et al. [1,2], which consists of controversial discussions from debate portals[3] and ChangeMyView, and on which the argument search engine args.me [14] also bases its arguments. The lab's participants can choose between the downloadable corpus and args' API.[4] In our implementation we work with the downloadable dataset. Moreover, the lab provides 50 topics on different areas such as "*abortion*" or "*gay marriage*", for which the lab participants have to find strong arguments from the provided dataset.

2 Our Approach: Concept and Implementation

In this section we give a short overview of our approach. First, we introduce the general concept, then we discuss the preprocessing of the provided data as well as another dataset that we use to estimate the convincingness of premises. Finally, we show how we find "strong" arguments.

2.1 Concept

We follow the principles developed in [6,8], which we summarize here briefly. First the set of claims $\mathcal{C} = \{c_1, c_2, \dots\}$ of the collection is clustered such that claims with the same meaning are assigned to the same claim cluster, yielding $\Gamma = \{\gamma_1, \gamma_2, \dots\}$ with $\gamma_i, \gamma_j \subseteq \mathcal{C}$ and $\gamma_i \cap \gamma_j = \emptyset$ in an offline operation (see Section 2.2). Note that the claims in args.me are sometimes formulated as questions or topic titles. However, in the following we will only refer to claims. We precluster the claims because often, the same claim appears in different formulations in a large collection, and we want to consider all variants of the same claim at the same time. This is even more important for

[3] The arguments were extracted from the following debate portals: `debatewise.org`, `idebate.org`, `debatepedia.org`, and `debate.org`.

[4] args' API: `https://www.args.me/api-en.html`. The downloadable dataset of args: `https://zenodo.org/record/3734893#.Xw24QCgzaUk`.

premises, since premises with the same meaning, but different formulations are often used in many claims, but we want to retrieve that premise only once. We thus cluster the set of premises $\mathcal{P} = \{p_1, p_2, \ldots\}$ and a set $\mathit{\Pi} = \{\pi_1, \pi_2, \ldots\}$ with $\pi_i, \pi_j \subseteq \mathcal{P}$ and $\pi_i \cap \pi_j = \emptyset$ of premise clusters is formed such that premises with the same meaning are put into the same premise cluster.

Now, given a *query claim* q, we apply a two-step retrieval approach. In the first step, our approach locates the claims in $\mathcal{C}$ that are most similar to q, following the observation that the more similar a claim is to a query, the more relevant are the premises of that claim to the query [7]. In the second step, we locate the claim clusters containing these claims, collect all premises related to a claim of one of these claim clusters, and finally determine the premise clusters to which these premises belong. For the ranking of premise clusters, we now apply a probabilistic ranking method. Thus, our goal is to compute $P(\pi_j|q)$, that is, the probability that π_j is chosen as supporting or attacking premise cluster for q. We can calculate $P(\pi_j|q)$ by iterating over all premises in π_j and aggregating their individual probabilities, i.e., $P(\pi_j|q) = \sum_{p \in \pi_j} P(p|q)$, where $P(p|q)$ is the probability that p is chosen as support or attack for q. $P(p|q)$ is defined by combining the two aforementioned steps, i.e., $P(p|q) = P(c|q) \cdot P(p|c)$. Formally, $P(c|q)$ denotes the probability that c is chosen as a similar claim to q. $P(p|c)$ denotes the probability that p is chosen as support (or attack) for c.

In our previous work [6] we estimated $P(p|c)$ exclusively via frequencies of premises. For our submissions to Task 1, we use a different approach [8] that also takes some dimensions of argument quality into account. We describe this approach in the following subsections.

2.2 Preprocessing of the Provided Dataset

Since the provided dataset by Ajjour et al. [1,2] originally consists of arguments with two components, that is, one claim with exactly one premise, we initially grouped all premises by their textually equal claim. Afterwards this grouping will become important because we calculate the convincingness of a premise in comparison to other premises of the same claim.

Then the contextualized embeddings of both the claims and the premises were derived by using Sentence-BERT (SBERT) [12].[5] For the clustering of claims as well as premises we followed the approach of our prior work [6] and implemented an agglomerative clustering applying Euclidean distance, the average linkage method, and a dynamic tree cut [11].[6]

[5] The framework used for this is available on `https://github.com/UKPLab/sentence-transformers`. The model we used for calculating the embeddings is "`roberta-large-nli-stsb-mean-tokens`", yielding embeddings of 1,024 dimensions each.

[6] For the agglomerative clustering we used the scripting language R and the packages STATS and FASTCLUSTER.

2.3 Including another Dataset to Estimate the Convincingness of Premises

Wachsmuth et al. [15] provide a dataset in which three experts assessed 320 arguments with respect to 15 argument quality dimensions. The arguments are distributed over 32 issue-stance pairs, i.e., 16 topics with two polarities and 10 premises each. Among these 15 dimensions there are the three main dimensions:

(1) logical quality in terms of the *cogency* or strength of an argument,
(2) rhetorical quality in terms of the persuasive effect of an argument or argumentation (called *effectiveness*), and
(3) dialectical quality in terms of the *reasonableness* of argumentation for resolving issues.

We considered the mean assessment values for the three main dimensions *cogency*, *reasonableness* and *effectiveness* and integrated the idea of Habernal and Gurevych [9] by deriving all combinations of (premise_1, premise_2) pairs with premises from the same issue-stance pairs and labels "1" or "2", whereby the labels signal which premise has a higher score with respect to a dimension. Pairs with equal mean value were omitted. Then, for the two premises of each pair as well as the corresponding (issue,stance) pair, we derived their SBERT embeddings, processed them to vectors consisting of the embeddings of the two premises each with the pointwise sum, difference, and product to the embedding of the corresponding (issue,stance) pair, yielding a vector of 6,144 dimension per (premise_1, premise_2) pair.[7] Then, we tested standard classifiers such as gradient boosting, logistic regression, or random forest with 32-fold-cross-validation and found that random forest performs best for cogency and effectiveness. For reasonableness Logistic Regression performed only slightly better. Using these best classifiers per dimension, we were able to precalculate the *dimension convincing frequencies* (DCFs) of the premises in the datset by Ajjour et al. [1,2]. Here, the DCF of a premise of a claim is calculated as the count how often the premise was better than the other premises belonging to the same claim in a cross comparison.

Now, both claims and premises could be indexed in two separate inverted indexes with the cluster and DCF information. We used Apache Lucene to build the indexes.[8]

2.4 Finding Strong Arguments

For each of the 50 topics which we regard as queries, we started by finding the most similar claims (*result claims*) using Divergence from Randomness (DFR) [3], because our previous work [7] implies that DFR is well suited for this task. Then all premises belonging to claims that are in the same cluster as the result claims were localized. The set of premises was then expanded with the set of premises in the same premise clusters, yielding the set of *result premises*.

Before calculating the premise cluster scores, first the premises were ranked individually. The ranking of these consists of the two components (1) similarity of the query

[7] The input can be determined by the elementwise computed Cartesian product of the embeddings of the following three sets, in compliance with the below order. The difference is positive. $\{\texttt{premise}_1, \texttt{premise}_2\}$, $\{+, -, *\}$, {(issue,stance)-pair}.

[8] https://lucene.apache.org/.

to the claim and (2) the sum of the three different DCFs per premise (see Section 2.3). Both (1) and (2) were normalized to have values between 0 and 1, allowing to use them like probabilities ($P(p|c)$ in the description above). The cluster scores were determined by aggregating the scores of the individual premises of the same cluster. From each cluster, only one representative was selected; in our implementation this is the longest premise as we followed the intuition that a longer premise is also more specific and therefore may be better suited as a representative.

As trec_eval sorts documents by the score values and not by rank values, it is important to handle tied scores. Furthermore, it is the score (integer or floating point) that is relevant for the TREC evaluation in the ranking. Therefore, the representatives were sorted in descending order by cluster score, then by length to break ties, and alphabetically if also the length was the same. To reflect this in the ranking, of all representatives with the same initial score, the scores were increased by the smallest possible delta in Java (10^{-17}) starting from the premise at the bottom.

Subsequent Adjustions We manually reviewed the results of our retrieval at a cutoff value of 30 and found that premises with less than 30 characters were usually completely useless as they are too unspecific or nonsense, so we removed them from the results.

3 Conclusion

In this paper we outlined our contribution (team Don Quixote) to the CLEF lab Touché. First we cluster claims and premises in an offline operation by their meaning. For a given query, we then work with a two-step retrieval process that first finds all similar claims and then, using the clusters, finds the relevant premises. For the ranking, we then calculate (1) the similarity of claim and query, and (2) the frequency with which a premise is more convincing than other relevant premises with respect to the three main argument quality dimensions cogency, reasonableness, and effectiveness. Describing (1) and (2) as probabilities, a ranking can be generated via their product. The code will be made available shortly.

Acknowledgements

This work has been funded by the Deutsche Forschungsgemeinschaft (DFG) within the project ReCAP, Grant Number 375342983 - 2018-2020, as part of the Priority Program "Robust Argumentation Machines (RATIO)" (SPP-1999).

References

1. Ajjour, Y., Wachsmuth, H., Kiesel, D., Riehmann, P., Fan, F., Castiglia, G., Adejoh, R., Fröhlich, B., Stein, B.: Visualization of the topic space of argument search results in args.me. In: Blanco, E., Lu, W. (eds.) EMNLP. pp. 60–65. Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/d18-2011, https://doi.org/10.18653/v1/d18-2011

2. Ajjour, Y., Wachsmuth, H., Kiesel, J., Potthast, M., Hagen, M., Stein, B.: Data acquisition for argument search: The args.me corpus. In: Benzmüller, C., Stuckenschmidt, H. (eds.) KI. Lecture Notes in Computer Science, vol. 11793, pp. 48–59. Springer (2019). https://doi.org/10.1007/978-3-030-30179-8_4, https://doi.org/10.1007/978-3-030-30179-8_4
3. Amati, G., van Rijsbergen, C.J.: Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Transactions on Information Systems **20**(4), 357–389 (2002). https://doi.org/10.1145/582415.582416
4. Bondarenko, A., Fröbe, M., Beloucif, M., Gienapp, L., Ajjour, Y., Panchenko, A., Biemann, C., Stein, B., Wachsmuth, H., Potthast, M., Hagen, M.: Overview of Touché 2020: Argument Retrieval. In: Working Notes Papers of the CLEF 2020 Evaluation Labs (Sep 2020)
5. Bondarenko, A., Hagen, M., Potthast, M., Wachsmuth, H., Beloucif, M., Biemann, C., Panchenko, A., Stein, B.: Touché: First shared task on argument retrieval. In: ECIR. Lecture Notes in Computer Science, vol. 12036, pp. 517–523. Springer (2020). https://doi.org/10.1007/978-3-030-45442-5_67, https://doi.org/10.1007/978-3-030-45442-5_67
6. Dumani, L., Neumann, P.J., Schenkel, R.: A framework for argument retrieval - ranking argument clusters by frequency and specificity. In: ECIR. Lecture Notes in Computer Science, vol. 12035, pp. 431–445. Springer (2020). https://doi.org/10.1007/978-3-030-45439-5_29, https://doi.org/10.1007/978-3-030-45439-5_29
7. Dumani, L., Schenkel, R.: A systematic comparison of methods for finding good premises for claims. In: SIGIR. pp. 957–960 (2019), https://doi.org/10.1145/3331184.3331282
8. Dumani, L., Schenkel, R.: Quality-aware ranking of arguments. In: CIKM (2020), accepted
9. Habernal, I., Gurevych, I.: Which argument is more convincing? Analyzing and predicting convincingness of web arguments using bidirectional LSTM. In: ACL (2016), https://www.aclweb.org/anthology/P16-1150/
10. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems **20**(4), 422–446 (2002). https://doi.org/10.1145/582415.582418, http://doi.acm.org/10.1145/582415.582418
11. Langfelder, P., Zhang, B., Horvath, S.: Dynamic tree cut: In-depth description, tests and applications (2009), https://horvath.genetics.ucla.edu/html/CoexpressionNetwork/BranchCutting/Supplement.pdf
12. Reimers, N., Schiller, B., Beck, T., Daxenberger, J., Stab, C., Gurevych, I.: Classification and clustering of arguments with contextualized word embeddings. In: ACL. pp. 567–578 (2019), https://www.aclweb.org/anthology/P19-1054/
13. Stede, M., Afantenos, S.D., Peldszus, A., Asher, N., Perret, J.: Parallel discourse annotations on a corpus of short texts. In: LREC (2016), http://www.lrec-conf.org/proceedings/lrec2016/summaries/477.html
14. Wachsmuth, H., Potthast, M., Khatib, K.A., Ajjour, Y., Puschmann, J., Qu, J., Dorsch, J., Morari, V., Bevendorff, J., Stein, B.: Building an argument search engine for the web. In: ArgMining@EMNLP. pp. 49–59. Association for Computational Linguistics (2017). https://doi.org/10.18653/v1/w17-5106, https://doi.org/10.18653/v1/w17-5106
15. Wachsmuth, H., Stein, B., Hirst, G., Prabhakaran, V., Bilu, Y., Hou, Y., Naderi, N., Alberdingk Thijm, T.: Computational argumentation quality assessment in natural language. In: EACL. pp. 176–187 (2017), https://aclweb.org/anthology/E17-1017/