Good Premises Retrieval via a Two-Stage Argument Retrieval Model

Lorik Dumani
Trier University
dumani@uni-trier.de

ABSTRACT

Computational argumentation is an emerging research area. An argument consists of a claim that is supported or attacked by at least one premise. Its intention is the persuasion of others to a certain standpoint. An important problem in this field is the retrieval of good premises for a given claim from a corpus of arguments. Given a claim, a first step of existing approaches is often to find other claims that are textually similar. Then, the similar claim's premises can be retrieved. This paper presents a research plan for an implementation of a two-stage argument retrieval model that first finds similar claims for a given query claim and then in the next step retrieves clusters of similar premises in a ranked order.

1. INTRODUCTION

Argumentation exists probably as long as humans communicate but research on computational argumentation has only recently become popular. In its simplest case an *argument* consists of a *claim* or a standpoint that is supported or attacked by at least one *premise* [10]. These relations between claims and premises can be expressed by argument graphs. The purpose of argumentation is the persuasion of others towards a certain standpoint. Since premises can in turn be attacked or supported, often large argument networks emerge for a major claim [10].

Our ultimate goal is, to support users arguing for or against a topic by providing the best premises to similar topics in a ranked order by convincingness, trustworthiness or user context. There already exist argument search engines like ARGS[1] or ARGUMENTEXT[2] that take a claim as input and return a list of premises that support or attack the query claim. These systems usually work on precomputed argument graphs that were either mined from texts or extracted from dedicated argument websites like `idebate.org` or `debatewise.org`. One challenge in premises retrieval is the small textual overlap between query claim and good premises supporting or attacking. In this paper we present a two-stage argument retrieval model. In contrast to existing methods like [15] which often use a combination of claim and premise as a retrieval unit, we argue that a more promising and principled approach than directly querying for premises is a two-stage process that first retrieves, given a query claim, matching claims from the argument collection, and then considers their premises only. Then, instead of retrieving single premises we aim to cluster similar premises and to retrieve ranked clusters of premises.

For the remainder of this paper Section 2 provides an overview of fundamentals such as an introduction to the related project ReCAP, and the common definition of arguments and argumentation. In Section 3 we present our research plan to retrieve clusters of premises for a query claim. Section 4 describes our evaluation plan and Section 5 serves with some results we found. Section 6 provides an overview of related work and Section 7 concludes the paper with some future works.

2. FUNDAMENTALS

This section introduces this work's related project ReCAP as well as the common definition of arguments and argumentation.

2.1 Project Context

This work is part of the ReCAP project described in [1] which is part of the DFG priority program robust argumentation machines (RATIO)[3].

ReCAP is an acronym for Information Retrieval and Case-Based Reasoning for Robust Deliberation and Synthesis of Arguments in the Political Discourse. The ReCAP project follows the vision of future argumentation machines that support researchers, journalistic writers, as well as human decision makers to obtain a comprehensive overview of current arguments and opinions related to a certain topic. Furthermore, it aims to develop personal and well-founded opinions that are justified by convincing arguments. While existing search engines are limited to achieve this approach, since they primarily operate on the textual level, such argumentation machines will reason on the knowledge level formed by arguments and argumentation structures. In [1] we propose a general architecture for an argumentation machine with focus on novel contributions to and confluence of me-

[1] `www.args.me`
[2] `www.argumentsearch.com`
[3] `www.spp-ratio.de`

31st GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), 11.06.2019 - 14.06.2019, Saarburg, Germany.

thods from Information Retrieval (IR) and Knowledge Representation and Reasoning (RI), in particular Case-Based Reasoning. *Deliberation* finds and weighs all arguments supporting or opposing some question or topic based on the available knowledge, e.g. by assessing their strength or factual correctness, to enable informed decision making, e.g. for a political action. *Synthesis* tries to generate new arguments for an upcoming topic based on transferring an existing relevant argument to the new topic and adapting it to the new environment.

This paper contributes to the retrieval of arguments, more specifically to the retrieval of clusters of the best premises in a ranked order for a given query claim from a corpus of arguments.

2.2 Argumentation

Argumentation is omnipresent and exists probably as long as humans communicate with each other and research on argumentation was already been studied by Aristotle more than 2,300 year ago [6]. By definition, an argument consists of a *claim* or standpoint supported or opposed by reasons or *premises* [10]. The terms claim and premise can be subsumed under the term *argument units* [3].

As shown in Figure 1 relations between claims and premises can be expressed by argument graphs. The main claim in a graph is called *major claim* [13] and since premises can in turn be attacked or supported, often large argument networks emerge for a major claim [10]. As Figure 1 suggests, an argument unit such as p_1 can also be used as a premise to support another claim.

In this example the premises support or attack the claim but the kind of support or attack is not further specified. However, supports can be specified with so-called inference schemes [17]. Those schemes are templates for argumentation that consist of claims and premises that are enriched with descriptors that assign different roles to different argument components to ease the choice of the correct scheme. Following [17], the support for the inference $p_1 \rightarrow C$ in this example can be specified as "*positive consequence*". The descriptor for the premise in this scheme is "*If A is brought about, good consequences will plausibly occur*". We can interpret a reduce in oil dependency as a good consequence. The descriptor for the claim in this scheme is "*A should be brought about*". The variable *A* in the descriptor can be replaced with the demand to build new nuclear plants. In contrast to supporting relations, there is no standard for the specification of attacking relations in argumentation theory yet.

Wachsmuth et al. provide in [16] a collection of approaches in literature to measure argument quality in natural language. Furthermore, they define a taxonomy of dimensions to measure. The dimensions of argument quality can be divided into the three dimensions *logical quality* in terms of the cogency or strength of an argument, *rhetorical quality* in terms of the persuasive effect of an argument or argumentation, and *dialectic quality* in terms of the reasonableness of argumentation for resolving issues [16].

3. RESEARCH PLAN

This section illustrates the research plan for implementing the two-stage retrieval system. We explain the necessity of the two stages and challenges we expect.

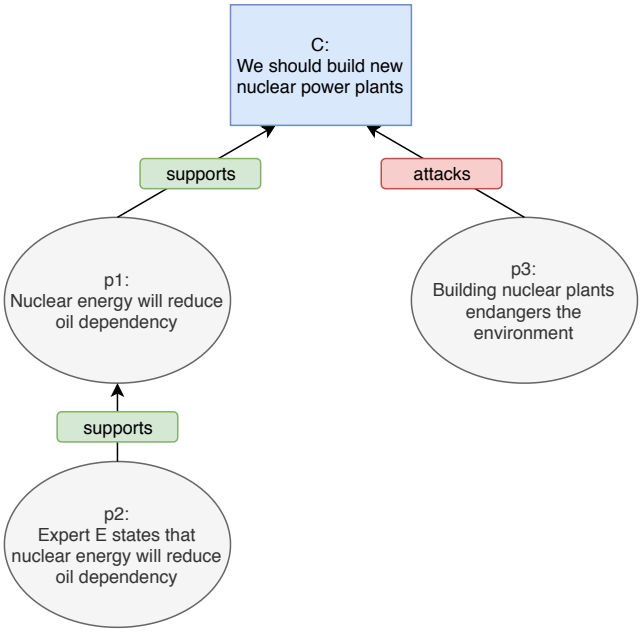

Figure 1: Simple argument graph showing the relations between argument units.

3.1 Two-stage Retrieval Process

Our ultimate goal is the retrieval of good premises supporting and attacking a given query claim or, more general, related to a query topic. Such a query could be a full sentence like e.g. "*Find arguments to abandon nuclear energy*" or just consist of relevant terms such as "*abandon nuclear energy*". One major challenge in the retrieval of premises is that a good, convincing, and related premise to the query does not necessarily need to have much textual overlap. This can be illustrated with the premise "*wind and solar energy can already provide most of the energy we need*" for the upper query claims. A less good premise could be "*I don't like nuclear energy. I would abandon it*". It is evident that the former premise only overlaps in the rather general term "*energy*" but is more convincing than the latter premise which however overlaps in the three words "*abandon*", "*nuclear*", and "*energy*".

Since arguments consist of claims and premises, the premises are directly tied to the claim, so we can tackle this problem by using a two-stage retrieval process that first retrieves, given a query claim, matching claims from the argument collection, and then considers their premises only. In the first step we only search for similar claims to the user's query claim, i.e., ignoring the premises at this point of time. Then in the second step we cluster similar premises and retrieve them in a ranked order.

3.2 The First Stage

In order to find relevant claims to a query claim we need to find claims that are semantically similar to the query claim. More precise, we need to find claims that have relevant premises to the query. So the challenge is to use basically syntactic similarity to achieve semantic similarity. In order to estimate the probability that a claim is relevant to the query, we can use any similarity measure we identify for textual

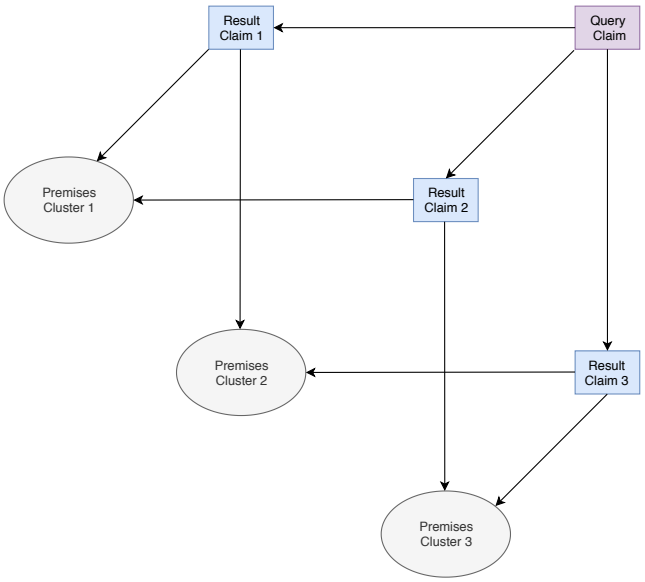

Figure 2: From a query to similar claims to clusters of premises.

data such as a plain language model, possibly with additional smoothing and taking the textual context of the claim into account.

3.3 The Second Stage

Since we are searching for good premises for a query claim that are obtained from similar claims to the query claim, we can assume that similar claims often have similar premises. Furthermore, as we are working with a large corpus of arguments, we will find a lot of similar premises, probably from semantically completely different claims. So instead of searching for single premises we group similar premises and search for clusters of premises. For clustering all premises we can first convert all premises with the same stance into embedding vectors and then perform a hierarchical clustering. Instead of computing own models to get embedding vectors we can make use of existing models such as the Universal Sentence Encoder described in [2]. We can use the Euclidean distance to compute distances between vectors. Clustering can be accomplished with agglomerative clustering, which is a bottom-up approach. Since we prefer smaller clusters to keep the number of false positives per cluster to a minimum, complete linkage is a good way to connect clusters [9].

Figure 2 visualizes an example of the relation between a query, similar claims, and clusters of similar premises. Here, we have to answer the research question how often premises which are similar to a premise do appear in claims that are similar to the query claim. In order to estimate the probability that a premise cluster should be chosen as supportive for a claim, we can use a simple approach as a frequency-styled argument, i.e., we need to count how frequently a premise cluster from this claim supports similar claims in a large corpus. Besides that, we can also consider to include inverse document frequency-styled arguments, i.e., we need to count how frequently the premise cluster was used as support or attack for other claims in a large corpus. Other legit approaches are to include estimates on truthfulness, appropriateness (of the premise for the claim), and confidence in expert. The ranking can incorporate factual correctness, convincingness, but also user context such as prior knowledge or belief in expert opinions, assumptions, and preferences. Therefore, we will include quality measures such as those described in [16]. However, we need to investigate in the strength of the cluster of premises. So far, there are only a few works in the early stages of development concerning the quality of single premises [16] but not clusters of premises.

3.4 Further Challenges

Another problem that should be paid attention to is the premise's stance, i.e., whether the premise supports or attacks the claim. But also the claim's stance needs to be determined. Consider e.g. the query claim "*Nuclear energy should be abolished*" and the claim "*Nuclear energy should not be abolished*". These claims take different views but have a high textual similarity which is why probably many retrieval methods would output a high similarity. Still the premises can not be adopted automatically. Moreover, claims often do not have a stance if they are queries like "*Should nuclear energy be extended?*" or consist only of terms like "*Nuclear energy*". One legit possibility for claims with neutral stances is to treat them as implicitly positive. Then, if a query claim and a result claim have the same stance, a premise that supports the result claim also supports the query claim whereas if the query claim and the result claim have opposite stances, a premise that supports the claim will attack the query claim and vice versa. Another approach that could make sense is to normalize stances of claims, i.e., to try to have only "positive" claims. Alternatively we could revert support and attack for negative claims. Still, that could be difficult if stance is not fully clear. Nevertheless, there exist algorithms for stance detection [12] which we can then use for this purpose.

Consider Figure 1 again. As already stated a claim can be used as premise to support or attack another claim. In this instance, the premise p_2 "*Expert E states that nuclear energy will reduce oil dependency*" is used to support the argument unit p_1 "*Nuclear energy will reduce oil dependency*" which in turn is used as premise to support the claim C "*We should build new nuclear power plants*". We need to investigate in the transitivity of inferences. In the example in Figure 1 to which extent e.g. p_2 is supportive for C. Analogously to that we need to investigate in the case whether a premise is supportive to a claim if the premise attacks another premise that in turn attacks the claim. Assume there would be a premise p_4 "*Humans endanger the environment either way*" that attacks premise p_3 "*Building nuclear plants endangers the environment*" which in turn already attacks claim C in Figure 1. So we want to examine how supportive premises such as p_4 are generally to a claim. In [15] Wachsmuth et al. simply adopt these as own premises for the claim. However, we will investigate whether a partial score or a damping factor yields better results. Since we are working with clusters of premises we can select one premise as representative. This could, for example, be the premise most similar to the centroid in the cluster. Please remember that premises are converted to embedding vectors to compute the clusters.

So far we have considered less complex queries such as "*what are good reasons for nuclear energy*". A query however can be much more complicated e.g. by the use of constraints. Such a more complex query could be "*what are common statements with factual evidences of Expert E in the*

last three months that nuclear energy is a viable option in Germany". In this example a user demands factual evidences for a geographically restricted area of a certain expert for a certain topic in a certain time span. Furthermore, the context could be desired to be restricted to opinions by certain interest groups or parties with certain political orientation such as left-wing parties. An approach could be to divide complex queries into sub queries. If the query is expressed as a coherent sentence its tree can be derived by the use of Part-of-Speech implementations such as [14]. Then, a cut can deliver useful sub queries.

4. EVALUATION PLAN

Instead of creating argument collections which is a very time consuming task or automatically mine arguments from natural language texts which might be noisy we will adapt the idea of [15] and make use of several debate portals. In fact we use `idebate.org`, `debatewise.org`, `debatepedia.org`, and `debate.org` as starting point. While the first three are of high quality, the latter is of lower quality, i.e., some few premises consist of insults or nonsense. However, the latter contains much more debates as the other three together. We expect this constellation to result in good diversification. The constructions of debate portals already serve with argument structures. One questioner asks the community about a topic, e.g., "*Should we build new nuclear power plants*". Then users of the community can directly answer the questions and substantiate their posts e.g. with facts or examples. Many debate portals also provide the possibility of adding a stance for or against to an answer, as do the portals we have selected for our study. The main advantages of debate portals are that the posts are not artificial but close to reality. Besides that they are coherent. Following [15] we use the debate portals' queries as claims and their answers as premises to build arguments.

We can divide the evaluation of the two-stage retrieval process into two evaluation steps. First we want to find similar claims to a query claim. This can be achieved via an existing textual similarity method. In order to decide which similarity method is suitable we can take a small number n of query claims and build pools of depth k by a union of result claims of existing similarity methods. Then, annotators can manually assess the similarity of each (query claim, result claim) pair e.g. in the range between 1 (nothing in common) and 5 (semantically equal). The question which similarity method should be adopted for the retrieval of claims can be shifted to the question which method's ranking comes closest to the annotations. We will use state-of-the-art ranking measures such as nDCG [8] for the evaluation of rankings.

After we determined the most similar claims to a query claim we want to retrieve their directly tied (clusters of) premises. In order to validate the hypothesis that claims highly similar to the query claim also have premises that are highly relevant for the query claim, we can take a fix number of (query claim, result claim, result claim premise) pairs of different similarities and let annotators manually assess the pairs e.g. on a binary scale where the annotators are not aware of the actual result claim. The higher the similarity is between two claims the more relevant the one's premises should be to the other claim.

Furthermore, we need an end-to-end analysis to evaluate the overall performance of our premise retrieval approach, i.e., how well can our approach retrieve premises for a given claim. For a subset of our query claims, we will build a pool of all result premises in the top-k (for some $k \in \mathbb{N}$) of all result lists and let annotators assess the premises' relevance as explained above. In addition to that, we can conduct a user study with more participants to overcome possible shortcoming of having only few annotators to check the results. By the use of nDCG at different cutoffs, averaged over all queries, we can evaluate different retrieval methods for this end-to-end analysis.

5. PRELIMINARY RESULTS

In this section we give an overview of results we found so far by investigating the stages of the two-stage retrieval model. First we describe how we built our dataset consisting of arguments, then we describe the first, and then the second step of the two-step retrieval process.

The dataset described in [15] is not publicly available, therefore we reconstructed a similar dataset following the approach in that paper. We crawled the arguments from four debate portals, namely `debate.org`, `debatepedia.org`, `debatewise.org`, and `idebate.org`. After the arguments were extracted, they were indexed with Apache Lucene. In the end, this resulted in overall 59,126 claims with 695,818 premises, so on average about 11.8 premises per claim.

We now describe the first step of the two-step retrieval process. Since real-life query inputs of users are difficult to find, we drew a random sample of 233 claims and used them as queries. In order to avoid claims that address completely random topics, our sample contained only claims that are related to the topic "energy". To do so, we trained a word-embedding-model on the 59,126 claims of our corpus using DeepLearning4j[4]. Then, we retrieved the nearest words of the word energy and filtered out inappropriate suggestions. Inappropriate suggestions were those that had nothing in common with our topic energy in the broadest sense. We repeated this approach five times for all newly added suggestions. In the end, we obtained 44 words such as "*nuclear*", "*electricity*", "*wind*", "*solar*", "*oil*", "*emission*", etc. We got 1,529 candidate claims where at least one of these words occurred, from which we drew a random sample of 233 claims, making sure by manual inspection that they are really related to the topic energy. To ensure that we end up with at least 200 valid claims, we have added another 33. In the end, we removed one claim because it appeared twice. We considered 196 different retrieval methods[5] implemented in Apache Lucene and retrieved, for each method, result claims for our 232 query claims. From the results, we built pools of depth 5, i.e., including any claim that appeared in the result list of any method at rank 5 or better. This resulted in 5171 (query claim, result claim) pairs. Please note, that pairs where the result claim was equal to the query claim are already excluded.

[4]Among others we used SkipGram as learning algorithm, the maximum window size was 8, the word vector size was 1000, the text was not preprocessed, and the number of iterations over the whole corpus was 15.

[5]Apache Lucene (Version 7.6.0) provides 139 similarity methods as well as a class for multiple similarities. We tested all combinations of the best methods' variants of Divergence from Randomness, Divergence from Independence, information-based models, and Axiomatic approaches as well as BM25 and Jelinek-Mercer in a first run and got $\sum_{k=2}^{6} \binom{6}{k} = 57$ new methods, resulting in 196 methods.

The user-perceived similarity of each (query claim, result claim) pair was independently assessed by at least two annotators on the scale from 1 to 5. A total of eight people participated in the annotation. They are all included in the ReCAP project and were introduced to the basics of argumentation theory. Table 1 explains the meanings of the different levels. The underlying assumption of this scale is that all premises of claims rated 4 or 5 should apply to the query claim, whereas no premises of claims rated 1 should apply. For claims rated 3, we expect that a good number of premises match, whereas premises of claims rated 2 would only rarely match. The annotators were confronted with the query claim and a result claim and were asked to assess how well they expect the premises of the result claim (that were unknown to them) would match the query claim. Since we only wanted to measure the relevance of claims at this point, the actual premises were not considered at this point, but investigated later. Since polarity of premises is not in the focus of this study, we collapse the levels 4 and 5 into a single level 4 for this study. As every pair of query claim and result claim was assessed by at least two annotators, the final relevance value of a result claim for a query claim was computed as the mean value of the corresponding assessments.

Using the assessed pool of results as a gold standard, we evaluated the performance of the 196 retrieval methods under consideration for the claim retrieval task, using nDCG@k [8] with cutoff values $k \in \{1, 2, 5\}$ as quality metric. Our results clearly show that the BM25 [11] scoring method used in previous works is usually not a good choice, especially for cutoff 5, which is a realistic cutoff for a system that aims at finding the top-10 premises. In contrast to the method Divergence from Randomness (DFR) [7], which yielded an nDCG@5 of 0.7982, BM25 yielded only 0.7616.

We now focus on the second step of the two-stage retrieval framework, retrieving the premises of claims similar to the query claim. Our goal here is to verify the assumption made above that claims highly similar to the query claim also have premises that are highly relevant for the query claim. To systematically approach this question, we formed triples of the form (query claim, result claim, result premise) from the above-mentioned pool, where the result premise is a premise of the result claim. We grouped the triples according to the relevance of the result claim to the query claim, forming groups of the relevance ranges $[n, n + 0.5)$ for $n \in \{n : 1 \leq n \leq 3.5\}$ and $[4, 4]$, which yielded seven groups. Then, we randomly drew 100 (query claim, result claim, result premise) triples from each group and had two annotators manually assess the relevance of the result premise for the query claim (without seeing the result claim), resulting in 1400 assessments. Annotators could choose between either *not relevant* or *relevant* with three different stances: *query with neutral stance*, *premise with same stance as query* and *premise with opposite stance as query*. As we did with claims before, we ignore the stances of premises since we only want to focus on their relevance, and many claims of our dataset do not have a stance anyway. We thus consider only binary relevance for premises from now on. Our preliminary results support the observation that the more relevant a claim for the query is, the more relevant premises it yields. For example, 80 % of the premises of the result claim in interval $[4, 4]$ were relevant to the query claim. In comparison, only 6 % of the premises in interval $[1, 1.5]$ were relevant to the query claim. So if a search engine performs well at the claim retrieval task, it should also perform well at the subsequent premise retrieval task; the initial hypothesis is thus validated.

Table 1: Relevance levels for claim assessment

score	meaning
5	The claims are equal.
4	The claims differ in polarity, but are otherwise equal.
3	The claims differ in specificity or extent.
2	The claims address the same topic, but are unrelated.
1	The claims are unrelated.

6. RELATED WORK

Wachsmuth et al. [15] introduce one of the first prototypes of an argument search engine called ARGS. Their system operates on arguments crawled from debate portals. Given a user query, the system retrieves, ranks, and presents premises supporting and attacking the query claim, taking similarity of the query claim with the premise, its corresponding claim, and other contextual information into account. They apply a standard BM25F ranking model implemented on top of Lucene. In contrast to their system, we did not restrict ourself to BM25 or variants, but evaluated 196 different similarity methods for claim retrieval.

Stab et al. [12] present ARGUMENTEXT, an argument retrieval system capable of retrieving topic-relevant sentential arguments from a large collection of diverse Web texts for any given controversial topic. The system first retrieves relevant documents, then it identifies arguments and classifies them as "pro" or "con", and presents them ranked by relevance in a web interface. In their implementation, they make use of Elasticsearch and BM25 to retrieve the top-ranked documents. In contrast to this work, we do not consider the argument mining task, but assume that we operate on a collection of arguments with claims and premises. However, in another work Habernal and Gurevych [4] propose a semi-supervised model for argumentation mining of user-generated Web content.

In [5], Habernal and Gurevych address the relevance of premises to estimate the convincingness of arguments using neural networks. Since relevance underlies a subjective judgement they first confronted users in a crowdsourced task with pairs of premises to decide which premise is more convincing, and then used a bidirectional LSTM to predict which argument is more convincing. Wachsmuth et al. [16] consider the problem of judging the relevance of arguments and provide an overview of the work on computational argumentation quality in natural language, including theories and approaches. Approaches that predict relevance or convincingness of premises can be useful to rank premises.

7. CONCLUSION AND FUTURE WORK

Retrieving good premises for claims is an important, but difficult problem for which no good solutions exist yet. This paper has provided some insights that a two-stage retrieval process that first retrieves claims, and then ranks their clustered premises can be a step towards a solution. The best premises are found for the most similar claims, according to assessments by human annotators, is already good. We showed that, instead of exhaustively assessing all retrieved

premises for a claim, it is sufficient to assess only the retrieved claims, which is an order of magnitude less work.

Our future work will include ranking methods for premises. We will also examine additional quality-based premise features [16] such as convincingness or correctness. We plan for a public Web application as an interface to our premise retrieval system.

We will also tackle the task to detect stances. Although debate portals ask users to add stances to the premises, these stances are related to the claim, but the claims' stances are not further specified. Hence, premises that support a claim may attack a claim with an opposite stance and vice versa.

8. ACKNOWLEDGMENTS

I would like to thank my supervisor Ralf Schenkel for his invaluable help in creating this paper.

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).

9. REFERENCES

[1] R. Bergmann, R. Schenkel, L. Dumani, and S. Ollinger. Recap - information retrieval and case-based reasoning for robust deliberation and synthesis of arguments in the political discourse. In *Proceedings of the Conference "Lernen, Wissen, Daten, Analysen", LWDA 2018, Mannheim, Germany, August 22-24, 2018.*, pages 49–60, 2018.

[2] D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, and R. Kurzweil. Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 169–174, 2018.

[3] J. Eckle-Kohler, R. Kluge, and I. Gurevych. On the role of discourse markers for discriminating claims and premises in argumentative discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 2236–2242, 2015.

[4] I. Habernal and I. Gurevych. Exploiting debate portals for semi-supervised argumentation mining in user-generated web discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 2127–2137, 2015.

[5] I. Habernal and I. Gurevych. Which argument is more convincing? analyzing and predicting convincingness of web arguments using bidirectional LSTM. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

[6] I. Habernal, R. Hannemann, C. Pollak, C. Klamm, P. Pauli, and I. Gurevych. Argotario: Computational argumentation meets serious games. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017 - System Demonstrations*, pages 7–12, 2017.

[7] S. P. Harter. A probabilistic approach to automatic keyword indexing. *JASIS*, 26(4):197–206, 1975.

[8] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[9] G. N. Lance and W. T. Williams. Mixed-data classificatory programs I - agglomerative systems. *Australian Computer Journal*, 1(1):15–20, 1967.

[10] A. Peldszus and M. Stede. From argument diagrams to argumentation mining in texts: A survey. *IJCINI*, 7(1):1–31, 2013.

[11] S. E. Robertson and H. Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

[12] C. Stab, J. Daxenberger, C. Stahlhut, T. Miller, B. Schiller, C. Tauchmann, S. Eger, and I. Gurevych. Argumentext: Searching for arguments in heterogeneous sources. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 2-4, 2018, Demonstrations*, pages 21–25, 2018.

[13] C. Stab, C. Kirschner, J. Eckle-Kohler, and I. Gurevych. Argumentation mining in persuasive essays and scientific articles from the discourse structure perspective. In *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing, Forlì-Cesena, Italy, July 21-25, 2014.*, 2014.

[14] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*, 2003.

[15] H. Wachsmuth, M. Potthast, K. A. Khatib, Y. Ajjour, J. Puschmann, J. Qu, J. Dorsch, V. Morari, J. Bevendorff, and B. Stein. Building an argument search engine for the web. In *Proceedings of the 4th Workshop on Argument Mining, ArgMining@EMNLP 2017, Copenhagen, Denmark, September 8, 2017*, pages 49–59, 2017.

[16] H. Wachsmuth, B. Stein, G. Hirst, V. Prabhakaran, Y. Bilu, Y. Hou, N. Naderi, and T. Alberdingk Thijm. 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, Valencia, Spain, April 3-7, 2017, Volume 1: Long Papers*, pages 176–187, 2017.

[17] D. Walton, C. Reed, and F. Macagno. *Argumentation Schemes*. Cambridge University Press, 2008.