Data Acquisition for Argument Search: The args.me Corpus

Yamen Ajjour[1](✉), Henning Wachsmuth[2], Johannes Kiesel[1], Martin Potthast[3], Matthias Hagen[4], and Benno Stein[1]

[1] Bauhaus-Universität Weimar, Weimar, Germany
{yamen.ajjour,johannes.kiesel,benno.stein}@uni-weimar.de
[2] Paderborn University, Paderborn, Germany
henningw@upb.de
[3] Leipzig University, Leipzig, Germany
martin.potthast@uni-leipzig.de
[4] Martin-Luther-Universität Halle-Wittenberg, Halle, Germany
matthias.hagen@informatik.uni-halle.de

Abstract. Argument search is the study of search engine technology that can retrieve arguments for potentially controversial topics or claims upon user request. The design of an argument search engine is tied to its underlying argument acquisition paradigm. More specifically, the employed paradigm controls the trade-off between retrieval precision and recall and thus determines basic search characteristics: Compiling an exhaustive argument corpus offline benefits precision at the expense of recall, whereas retrieving arguments from the web on-the-fly benefits recall at the expense of precision. This paper presents the new corpus of our argument search engine *args.me*, which follows the former paradigm. We freely provide the corpus to the community. With 387 606 arguments it is one of the largest argument resources available so far. In a qualitative analysis, we compare the args.me corpus acquisition paradigm to that of two other argument search engines, and we report first empirical insights into how people search with args.me.

1 Introduction

The web is rife with one-sided documents (marketing, lobbyism, propaganda, hyperpartisan news, etc.), but today's search engines are not well-equipped to deal with such kind of one-sidedness. Ignorant of the fact, they see documents as relevant that match a query's topic. For instance, if a user queries `feminism harms society`, a document that confirms this claim, all other things being equal, will be ranked higher than one denying it. Accordingly, preempting a conclusion on a controversial topic in a query will probably yield strongly biased results towards that conclusion, providing little opportunity to have one's beliefs challenged. Especially for controversial topics, a more nuanced approach may be advisable: *arguments* may be retrieved instead of (one-sided) documents enclosing them, and displayed alongside each other in a pro and con fashion towards a query's

C. Benzmüller and H. Stuckenschmidt (Eds.): KI 2019, LNAI 11793, pp. 48–59, 2019.
https://doi.org/10.1007/978-3-030-30179-8_4

claim. Technologies such as *IBM Debater* [10], *ArgumenText* [14], and our own argument search engine *args.me* [17] are the first such prototypes available. For these technologies, an argument consists of a conclusion together with supporting premises, e.g., "feminism did more good than harm" (conclusion), "since it has contributed a lot to gender equality" (premise).

A search engine typically implements an indexing process and a retrieval process [5]. In the context of argument search, the former acquires arguments (or argumentative documents), assesses their quality, and indexes them to facilitate the recurring execution of the retrieval process. The retrieval process, in turn, retrieves and ranks relevant arguments according to the users' queries [17].

The acquisition of arguments requires the availability of suitable sources, in particular sources which cover the whole range of topics that is of interest to the search engine's users. Depending on the argument acquisition paradigm employed, arguments must be mined from argumentative documents either at indexing time or at retrieval time. Most argument mining approaches are based on dedicated machine learning technology to extract arguments from text, trained on previously annotated corpora [3,11,15]. The training corpora available today consist exclusively of samples from specific text genres, such as news editorials, legal text, or student essays. This limits the sources that can be exploited for the still lacking generalizability of these approaches across domains [1,6].

Despite the fact that argument mining is still in its infancy and hence argument acquisition is limited, it is important to enable the study of the downstream search process. For the three aforementioned argument search engines, their authors pursue different solutions, each having their own advantages and disadvantages (see Sect. 2 for a qualitative analysis). While we introduced our argument search engine args.me and its underlying framework in previous work [17], the focus of this paper is the newly revised and extended argument corpus indexed by args.me, along with the acquisition paradigm it employs. Via distant supervision on dedicated online debate portals, we obtain big amounts of high-quality arguments for a wide range of topics with little to no development overhead. The altogether 387 606 arguments from 59 637 debates constitute one of the largest resources for computational argumentation available so far. We freely provide the complete corpus to the community.[1]

The paper is organized as follows. Section 2 presents background and related work on argument search engines, culminating in a qualitative analysis of three argument acquisition paradigms. Section 3 briefly illustrates the crawling of the debate portals covered by args.me as well as the employed distant supervision heuristics. Section 4 reports key statistics as well as distributions of arguments and debates in our corpus, and Sect. 5 overviews relevant computational argumentation tasks that can be tackled with the corpus. Based on a first log analysis, Sect. 6 provides insights into how people search with args.me.

[1] https://webis.de/data/args-me.html or https://doi.org/10.5281/zenodo.3274635.

2 Related Work

Computational argumentation research emanates from different domains and has been motivated by different applications. For example, artificial intelligence studies argumentative agents that persuade humans [13], computational linguistics studies argument mining in the context of writing support [15], and in the field of models for argumentation a web of arguments is envisioned with tools like the AIFdb to unify argument corpora to a standardized argument model [8]. While all these directions can also be relevant to retrieval scenarios, we focus on the specific challenges that argument search poses.

Argument search is a new research area centered around the idea of search engines that retrieve pro and con arguments for a given query. The typical steps include argument acquisition, argument indexing, argument quality assessment [10,14,17]. In the argument acquisition step, the task is to extract arguments from suitable sources, ensuring a wide topic coverage to be able to answer a wide variety of user queries. A key challenge in the acquisition step is to build a robust argument mining method tailored to specific argument sources—a recent study emphasized the difficulty of cross-domain argument mining [6].

The existing argument search prototypes [10,14,17] follow paradigmatically different approaches to argument acquisition: see Fig. 1 for a comparison. The choice of argument sources and mining methods is usually tightly coupled and constitutes a decisive step in designing an argument search engine. The smaller the ratio of explicit arguments to other text in the sources, the more effort needs to be invested to mine high-quality arguments.

ArgumenText (Fig. 1 bottom) follows web search engines in indexing entire web documents. Using a classifier trained on documents from multiple domains, ArgumenText then mines and ranks arguments from topically relevant documents at query time [16]. The advantages of this approach are recall maximization ("everything" is in the index) and the possibility to decide whether a text span is argumentative on a per-query basis. A disadvantage may arise from the aforementioned as of yet unsolved problem of cross-domain robustness [6].

IBM Debater's approach (Fig. 1 center) is to mine conclusions and premises of arguments from recognized sources (such as Wikipedia and high-reputation news portals) with classifiers trained for specific topics [9,10,12]. The arguments are indexed offline (i.e., unlike ArgumenText, the retrieval unit is an argument, not a document)—the complete documents may still be stored in an additional storage. Argument retrieval then boils down to topic filtering and ranking. While the source selection benefits argument quality, recall depends on the effort invested into the training of the classifiers (i.e., human labeling is involved to guarantee the effectiveness of the topic-specific classifiers).

Finally, the approach of args.me is shown in the top Fig. 1. Arguments from debate portals are indexed offline, similar to IBM Debater. However, instead of a classifier-based mining, we harvest arguments using distant supervision, exploiting the explicit debate structure provided by humans (including argument boundaries, pro and con stance, and meta data). This does not only benefit the retrieval precision, but also renders our approach agnostic to topics. A shortcom-

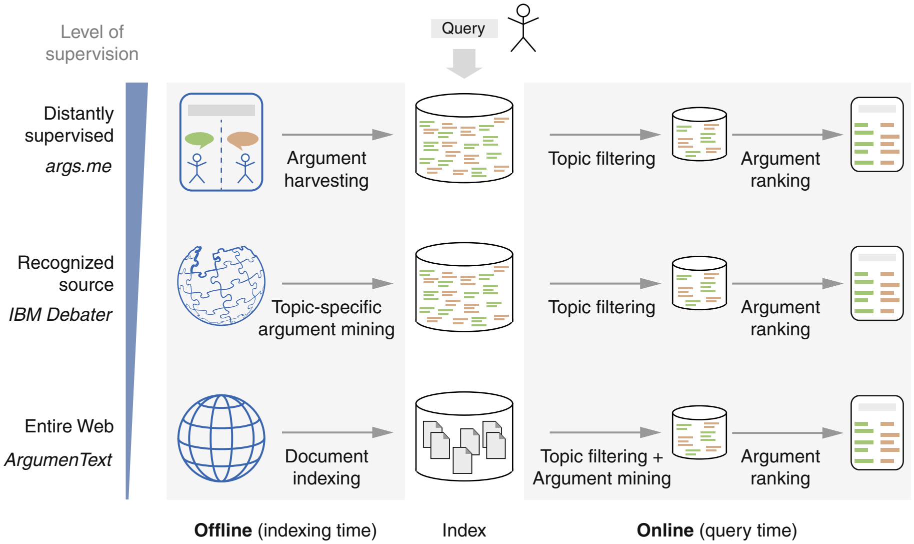

Fig. 1. Comparison of three general argument acquisition paradigms: args.me and IBM Debater index arguments offline, relying on distantly supervised harvesting and on mining from recognized sources respectively. ArgumenText indexes documents and mines online at query time. The *level of supervision* reflects the effort humans spent to create arguments from a source, which in turn implies notable differences regarding index sizes, topic bias, and noise in the data.

ing of our approach is that it needs to decide what is an argument at indexing time, independent of a query. To some extent, this restriction can be overcome in the future through more elaborated topic filtering and ranking algorithms. Besides, the gain of precision comes at the expense of recall as the number of sources qualifying for distantly-supervised argument harvesting is limited. In the next section, we briefly revisit the distant supervision heuristics of args.me underlying the extraction of arguments from debate portals [17].

3 Corpus Acquisition

Debate portals are websites dedicated to *organized* online debate. Not unlike debate clubs, users exchange arguments on controversial issues, allowing their audience to judge their merits. Some portals, such as debate.org, contain dialogical discussions, others, such as debatepedia.org, list arguments with pro and con stance for each covered topic. Both types of portals are largely balanced in terms of the number of pro and con arguments for each topic, allowing users to form opinions in an unbiased manner. Due to the wide range of covered topics and the high average argument quality, many debate portals are a valuable resource often used in computational argumentation research [2,4,7] and form the argument source of args.me [17].

Table 1. Example from the args.me corpus (context and meta information omitted for brevity).

Field	**Value**
Conclusion	Gay marriage
Stance	Pro
Premise	Gay marriage should be legal without a doubt. Marriage isn't about gender, it is about love. The reason why you marry someone is because you love them and you want to spend the rest of your life with them. Just because your significant other is the opposite gender as you doesn't mean anything, same if they are the same gender. Also, another thing to point out is religion versus gay marriage. Religion means beliefs. If you believe in something, it's what you think, not what should happen. Especially, some people who believe in God follow the bible and hence believe that gay marriage is immoral and disgusting. But just because a small group of people thinks this way, gay marriage shouldn't be outlawed.

In this work, we provide a corpus created from a new, revised crawl of debate portals covering arguments up to May 2019. As different events spark new debates, it is necessary for an argument search engine to provide up-to-date arguments. For args.me, we build software to automatically extract a list of all debate pages from the portals and to store these pages in the standard web archive format (WARC). These web archive files form the raw data for args.me's indexing pipeline. The debate portals contained in our corpus are (1) idebate.org, (2) debatepedia.org, (3) debatewise.org, and (4) debate.org.

As described by Wachsmuth et al. [17], we model an *argument* as a conclusion, a set of one or more premises, and a pro or con stance of each premise towards the conclusion. From each debate's page, we extract its arguments, the context they come from, and some meta information. The *context* of an argument is the text of the debate in which it was used, the title of the debate, and its URL. In terms of *meta information*, we generate a unique ID for each argument as well as a unique ID for the debate (based on the URL of the web page). We also extract the acquisition time of the debate for provenance. Table 1 shows an example of an argument in the args.me corpus.

Based on the structure of the debates, we developed portal-specific heuristics to extract the text of arguments. We briefly revisit these heuristics here, but refer the reader to the original publication for details [17]. A debate in dialogical portals consists mainly of a title and a sequence of argumentative posts by two opposing parties. In most cases, the title is a claim supported by a party (pro) and contested by the other (con). Heuristically, we consider the title to be the conclusion of an argument and each post to be a premise. The stance of the premise towards the conclusion corresponds to the position of the respective party in the debate. Monological portals require different heuristics. While the debate topics usually also are general claims (e.g., "abortion should be banned"),

Table 2. Statistics of the arguments in the args.me corpus, the arguments whose premise is pro and con towards the conclusion respectively, and the debates from each covered debate portal.

Debate Portal	Count of arguments	Count of pro stance	Count of con stance	Count of debates
Debatewise	14 353	8 514	5 839	693
IDebate.org	13 522	6 839	6 683	618
Debatepedia	21 197	15 791	5 406	236
Debate.org	338 534	168 955	169 579	58 090
$\sum$	**387 606**	200 099	187 507	59 637

the individual contributions to a debate should rather be seen as single arguments (i.e., a conclusion with a premise) organized as pro or con towards the debate's topic.

From the extracted arguments, we remove the ones with conclusions formulated as questions (to favor decisive arguments) and we remove commonplace phrases (e.g., "this house believes that" at the start of arguments).

4 The args.me Corpus

The output of the acquisition process above is the *args.me corpus*, which represents the data basis underlying our argument search engine. Table 2 shows the number of arguments and debates from each debate portal included in the corpus. As shown, debate.org is the dominant source among them, but the other three still add up to about 50 000 arguments in total. In general, pro arguments and con arguments are nearly balanced.

Conclusions can be supported or attacked by multiple arguments. The number of existing arguments in our corpus per conclusion gives a lower bound of the number of arguments that may be retrieved for an input conclusion. To obtain this bound, we grouped arguments that have the same conclusion. The average count of arguments per conclusion in the corpus amounts to 5.5. Figure 2a shows a histogram of the conclusions in our dataset using the count of arguments per conclusion. Most of the conclusions are directly addressed in 1 to 10 arguments, whereas only a few conclusions reach more than 20 arguments, the maximum being 2 838.

Our dataset contains around 60 000 debates to which the arguments have a pro or con stance. The average count of arguments per debate in our dataset amounts to 6.5. Figure 2b shows a histogram of the number of arguments over debates in the args.me corpus. Most debates include 6 to 10 arguments. Again, only a few debates reach more than 20 arguments.

Figures 2c and d show two histograms for the count of conclusions and premises over their length in tokens. As can be seen, there is much variance

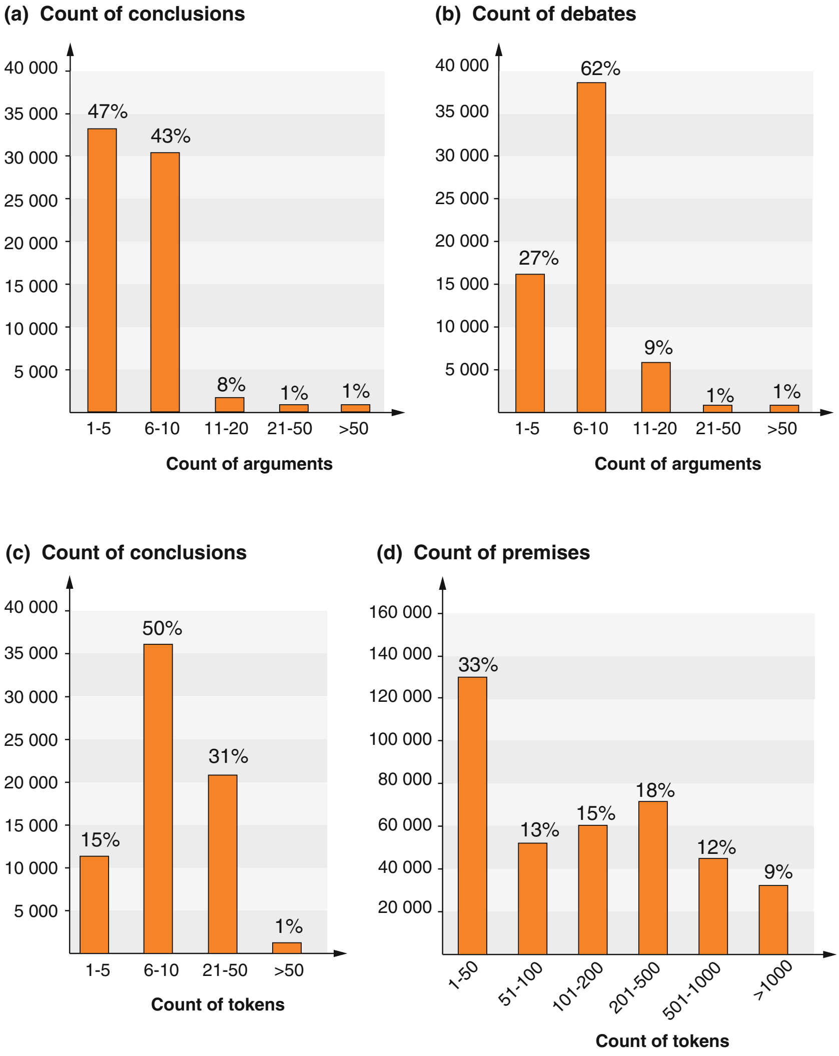

Fig. 2. Histograms illustrating key statistics of the args.me corpus. (a) The number of arguments over conclusions. (b) The number of arguments over debates. (c) The number of conclusions over the count of tokens. (d) The number of premises over the count of tokens.

in the length of both types of argument units. The mean length of conclusions in the corpus is 8.3 tokens, whereas the premises span 293 tokens on average. The high length of the premises in comparison to the conclusions suggests that some of them actually include multiple premises. Since a real argument unit segmentation algorithm is lacking in the args.me framework so far [1], we decided to leave all premises combined, avoiding noise from faulty segmentation.

Table 3. Argument search tasks enabled by the args.me corpus along with their input and output.

Task	**Input**	**Output**
Same-side classification	Argument pair	{“same”, “opposite”}
Stance classification	{Argument, Topic}	{“pro”, “con”}
Argument relation classification	Argument pair	{“support”, “attack”, “none”}
Argument conclusion generation	Premise	Conclusion

5 Argument Search Tasks

The args.me corpus is meant for studying multiple tasks relevant to argument search in particular, as well as to computational argumentation research in general. While some tasks should be performed online by an argument search engine, others can be performed offline to improve the quality of the corpus or to provide more information to the user. In what follows, we given a brief overview of the tasks for which approaches can be directly developed and evaluated using our corpus, for example, in a supervised machine learning setting. Table 3 lists these tasks along their input and output.

Same-Side Classification. Given two arguments on the same topic, decide whether they have the same or an opposite stance towards it. An argument search engine may address this task at indexing time to reduce noise: For example, if one argument has a clear, unambiguous stance towards a topic, the stance of others may be revised based on a comparison to that argument. Same-side classification can be studied on our corpus, since all its arguments comprise a stance towards their conclusion (i.e., its topic). Using the args.me corpus, we organized the same side stance classification challenge[2] with the goal of fostering the development of classifiers to perform the task.

Stance Classification. Given an argument along with a topic, classify whether the argument is pro or con towards the topic. An argument search engine may address this task online only, when given the topic in the form of a query. This is necessary in order to distinguish pro and con arguments so as to balance bias in the search results. Stance classification can be studied on our corpus similar to same-side classification; any approach to stance classification may also be used for same-side classification.

Argument Relation Classification. Given a pair of arguments, does one argument support or attack the other, or neither. An argument search engine may address this task offline, for instance, to identify counterarguments for a given arguments [18]. Argument relation classification can be studied on our corpus, since the corpus contains arguments whose conclusions represent premises in other arguments.

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

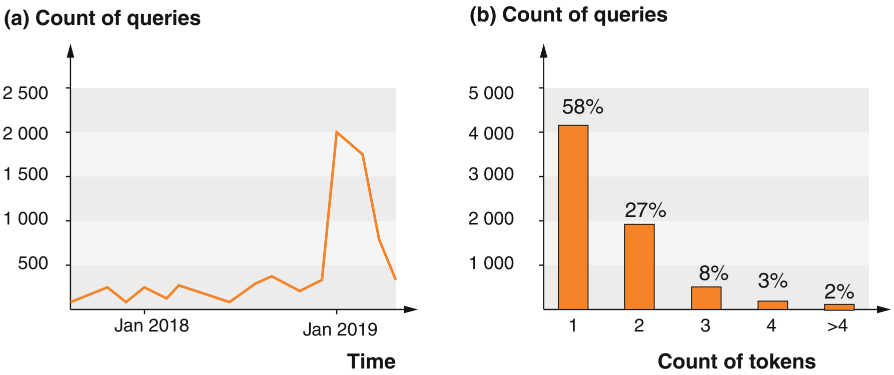

Fig. 3. Statistics of the queries sent to args.me between September 2017 and April 2019: (a) Plot of query distribution over time. (b) Histogram of the queries over their tokens count.

Argument Conclusion Generation. Given the premises of an argument, generate its conclusion. An argument search engine may address this task offline, in order to fill in missing conclusions not available at acquisition time, which may be the case if argument sources other than debate portals are included. Argument conclusion generation can be studied on our corpus, since each argument comes with both a premise and a conclusion.

Naturally, the corpus may also serve several other tasks related to argumentation, but may require additional labels for the arguments. Wachsmuth et al. [17] overview further argument search tasks.

6 First Insights from the args.me Query Log

In this section, we report on an analysis of the args.me query log to provide first insights into what users ask for when looking for arguments. The query log covers all queries that were posted to args.me between September 2017 and May 2019. So far, we assume args.me to be used by researchers mainly, hence the relatively small amount of about 13 000 queries in this period. In addition to the posted free text query, we store for each query an ID derived from the sender's IP address and the query time.

Before our analysis, we removed all queries that originated from our institutes to avoid confusing our analysis with test queries sent during development or presentations of args.me. We also removed all duplicate queries that were sent from the same sender within three seconds, resulting in 7084 queries. Figure 3a shows the distribution of the queries posted to args.me for each month in the covered period. On average, around 393 queries have been submitted per month by external people. The plot shows a peak at the beginning of 2019, where args.me was covered in German news media, suggesting a healthy interest in argument search.

Table 4. (a) Top ten queries found in the args.me query log, and (b) top ten conclusions of arguments in the args.me corpus, each with their absolute and relative frequencies.

(a) Query	**Absolute**	**Relative**
`climate change`	251	3.5%
`feminism`	193	2.7%
`abortion`	158	2.2%
`trump`	146	2.0%
`brexit`	128	1.8%
`death penalty`	73	1.0%
`google`	58	0.8%
`vegan`	57	0.8%
`nuclear energy`	56	0.8%
`donald trump`	47	0.7%

(b) Conclusion	**Absolute**	**Relative**
Abortion	2 838	0.7%
Gay Marriage	1 558	0.4%
Rap Battle	1 396	0.4%
Death Penalty	972	0.3%
God exists	790	0.2%
Gun control	719	0.1%
Ivf debate	432	0.1%
Animal testing	372	0.1%
I will not contradict myself	357	0.1%
Euthanasia	331	0.1%

The count of tokens in a query can be seen as an indicator of the specificity and complexity of user information needs. Short queries likely represent a topic, while long queries likely represent a claim or a conclusion. Figure 3b shows the distribution of the queries over their count of tokens. As shown, about 85% of the queries consist of two tokens at most. An example for a topic query is `abortion`, while a conclusion query may be `abortion should be banned`. Compared to conclusions which have a specific stance toward a topic, topic queries may indicate that a user seeks to overview both sides' arguments.

We analyzed topic queries sent to args.me in more detail. To identify unambiguous topic queries, we matched the queries in our log with a list of controversial topics extracted from Wikipedia.[3] We found that 20% of the topic queries exactly match one of the Wikipedia topics. The ten most frequently sent queries are listed in Table 4a, along with their absolute count and their relative occurrence among all queries. For comparison, Table 4b lists the ten most frequent conclusions of arguments in the args.me corpus. The comparison between the most frequent queries and conclusions shows some similarities and some divergence between the topics found in our corpus and those that people are interested in. In particular, the top ten queries mostly match controversial topics. Queries such as `donald trump`, `brexit`, and `global warming` are submitted often on args.me, but are not discussed that much in our corpus. Such queries indicate topics for which our corpus should be extended with arguments from other sources in the future.

7 Conclusion

Argument search is a research area that targets the retrieval of arguments (typically "pro" or "con") for queries on controversial topics. Though still in its

[3] https://en.wikipedia.org/wiki/Wikipedia:List_of_controversial_issues.

infancy, it has become clear that argument search engines provide a new and effective means to satisfy certain information needs. E.g., an argument search engine can help to compare and assess a user's standpoint since it contrasts both sides of a topic in a probably less biased manner. It can help to effectively close knowledge gaps, among others due do the succinct and concise form of arguments. With args.me, Wachsmuth et al. [17] present such a search engine, which is designed as a pipeline of modular tasks, integrating argument mining, argument matching, and argument ranking.

In this paper we focused on the first step of designing an argument search engine: the acquisition (mining) of arguments. This step includes the choice of argument sources as well as methods to extract the arguments from these sources. We compared the acquisition paradigm of args.me to those of IBM Debater [10] and ArgumenText [14]. The main difference between these approaches can be explained by the following two factors: (1) the level of supervision (high to low: distantly supervised/recognized source/unrestricted web), and (2) the point in time at which important processing steps are executed (offline, at indexing time/online, at query time). Due to the use of distant supervision, args.me can rather easily ensure a high average quality for the indexed arguments—which, however, comes at the price of a restricted recall, since the topics in args.me are limited to those found in debate portals.

We presented the corpus underlying args.me and freely release it for future research. With 387,606 arguments it is (to our knowledge) the currently largest argument resource available for computational argumentation research. Debate portals provide a balanced number of arguments with pro and con stance, a fact that helps to reduce bias in search results. We sketched four standard tasks that can be performed using our corpus and that should be tackled by an argument search engine. The analysis of arg.me's query log reveals that 20% of the queries match well-known controversial topics.

Future research on argument acquisition will focus on finding new argument sources along with tailored extraction methods for them. In this regard, social media and news portals appear promising to us, since they provide a wider and more recent topic coverage than debate portals. However, argument extraction methods for social media and news portals (either automatically or semi-automatically) are largely unexplored as of yet.

References

1. Ajjour, Y., Chen, W.F., Kiesel, J., Wachsmuth, H., Stein, B.: Unit segmentation of argumentative texts. In: Proceedings of the Fourth Workshop on Argument Mining (ArgMining 2017), pp. 118–128
2. Al-Khatib, K., Wachsmuth, H., Hagen, M., Köhler, J., Stein, B.: Cross-domain mining of argumentative text through distant supervision. In: Proceedings of the 15th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2016), pp. 1395–1404
3. Al-Khatib, K., Wachsmuth, H., Kiesel, J., Hagen, M., Stein, B.: A news editorial corpus for mining argumentation strategies. In: Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016), pp. 3433–3443

4. Cabrio, E., Villata, S.: Natural language arguments: A combined approach. In: Proceedings of the 20th European Conference on Artificial Intelligence (ECAI 2012), pp. 205–210
5. Croft, W.B., Metzler, D., Strohman, T.: Search Engines - Information Retrieval in Practice. Pearson Education, London (2009)
6. Daxenberger, J., Eger, S., Habernal, I., Stab, C., Gurevych, I.: What is the essence of a claim? Cross-domain claim identification. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), pp. 2055–2066
7. Habernal, I., Gurevych, I.: 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), pp. 1589–1599
8. Lawrence, J., Bex, F., Reed, C., Snaith, M.: AIFdb: Infrastructure for the argument web. In: Proceedings of the Fourth International Conference on Computational Models of Argument (COMMA 2012), pp. 215–516
9. Levy, R., Bilu, Y., Hershcovich, D., Aharoni, E., Slonim, N.: Context dependent claim detection. In: Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers (COLING 2014), pp. 1489–1500
10. Levy, R., Bogin, B., Gretz, S., Aharonov, R., Slonim, N.: Towards an argumentative content search engine using weak supervision. In: Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018), pp. 2066–2081
11. Moens, M.F., Boiy, E., Palau, R.M., Reed, C.: Automatic detection of arguments in legal texts. In: Proceedings of the 11th International Conference on Artificial Intelligence and Law (ICAIL 2007), pp. 225–230
12. Rinott, R., Dankin, L., Perez, C.A., Khapra, M.M., Aharoni, E., Slonim, N.: Show me your evidence – An automatic method for context dependent evidence detection. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015), pp. 440–450
13. Rosenfeld, A., Kraus, S.: Strategical argumentative agent for human persuasion. In: Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016), pp. 320–328
14. Stab, C., et al.: ArgumentText: searching for arguments in heterogeneous sources. In: Proceedings of 17th Annual Conference of North American Chapter of the Association for Computational Linguistics (NAACL 2018), pp. 2055–2066
15. Stab, C., Gurevych, I.: Identifying argumentative discourse structures in persuasive essays. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp. 46–56
16. Stab, C., Miller, T., Gurevych, I.: Cross-topic argument mining from heterogeneous sources using attention-based neural networks. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018), pp. 3664–3674
17. Wachsmuth, H., et al.: Building an argument search engine for the web. In: Proceedings of the Fourth Workshop on Argument Mining (ArgMining 2017), pp. 49–59
18. Wachsmuth, H., Syed, S., Stein, B.: Retrieval of the best counterargument without prior topic knowledge. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018), pp. 241–251