Building an Argument Search Engine for the Web

Henning Wachsmuth Martin Potthast Khalid Al-Khatib Yamen Ajjour
Jana Puschmann Jiani Qu Jonas Dorsch Viorel Morari Janek Bevendorff Benno Stein
Webis Group, Faculty of Media, Bauhaus-Universität Weimar, Germany
<firstname>.<lastname>@uni-weimar.de

Abstract

Computational argumentation is expected to play a critical role in the future of web search. To make this happen, many search-related questions must be revisited, such as how people query for arguments, how to mine arguments from the web, or how to rank them. In this paper, we develop an argument search framework for studying these and further questions. The framework allows for the composition of approaches to acquiring, mining, assessing, indexing, querying, retrieving, ranking, and presenting arguments while relying on standard infrastructure and interfaces. Based on the framework, we build a prototype search engine, called *args*, that relies on an initial, freely accessible index of nearly 300k arguments crawled from reliable web resources. The framework and the argument search engine are intended as an environment for collaborative research on computational argumentation and its practical evaluation.

1 Introduction

Web search has arrived at a high level of maturity, fulfilling many information needs on the first try. Today, leading search engines even answer factual queries directly, lifting the answers from relevant web pages (Pasca, 2011). However, as soon as there is not one single correct answer but many controversial opinions, getting an overview often takes long, since search engines offer little support. This is aggravated by what is now called fake news and alternative facts, requiring an assessment of the credibility of facts and their sources (Samadi et al., 2016). Computational argumentation is essential to improve the search experience in these regards.

The delivery of arguments for a given issue is seen as one of the main applications of computational argumentation (Rinott et al., 2015). Also, it plays an important role in others, such as automated decision making (Bench-Capon et al., 2009) and opinion summarization (Wang and Ling, 2016). Bex et al. (2013) presented a first search interface for a collection of argument resources, while recent work has tackled subtasks of argument search, such as mining arguments from web text (Habernal and Gurevych, 2015) and assessing their relevance (Wachsmuth et al., 2017b). Still, the actual search for arguments on the web remains largely unexplored (Section 2 summarizes the related work).

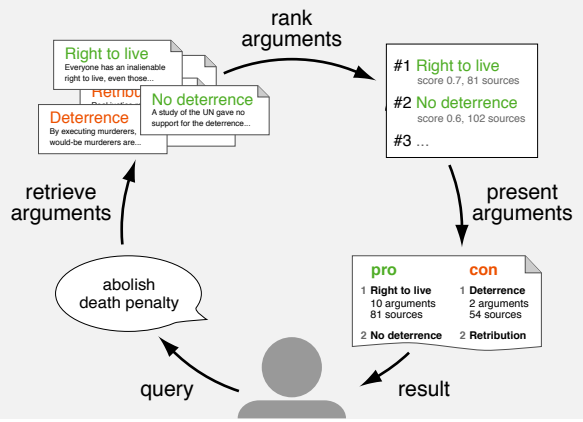

Figure 1: High-level view of the envisioned process of argument search from the user's perspective.

Figure 1 illustrates how an argument search process could look like. Several research questions arise in light of this process, starting from what information needs users have regarding arguments and how they query for them, over how to find arguments on the web, which of them to retrieve, and how to rank them, to how to present the arguments and how to interact with them.

This paper introduces a generic framework that we develop to study the mentioned and several further research questions related to argument search on the web. The framework pertains to the two

Proceedings of the 4th Workshop on Argument Mining, pages 49–59

main tasks of search engines, *indexing* and *retrieval* (Croft et al., 2009). The former covers the acquisition of candidate documents, the mining and assessment of arguments, and the actual indexing. The latter begins with the formulation of a search query, which triggers the retrieval and ranking of arguments, and it ends with the presentation of search results. The outlined steps are illustrated in Figure 2 and will be detailed in Section 3.

To achieve a wide proliferation and to foster collaborative research in the community, our framework implementation relies on standard technology. The argument model used represents the common ground of existing models, yet, in an extensible manner. Initially, we crawled and indexed a total of 291,440 arguments from five diverse online debate portals, exploiting the portals' structure to avoid mining errors and manual annotation while unifying the arguments based on the model (Section 4).

Given the framework and index, we created a prototype argument search engine, called *args*, that ranks arguments for any free text query (Section 5). *args* realizes the first argument search that runs on actual web content, but further research on argument mining, assessment, and similar is required to scale the index to large web crawls and to adapt the ranking to the specific properties of arguments. Our framework allows for doing so step by step, thereby providing a shared platform for shaping the future of web search and for evaluating approaches from computational argumentation in practice.

Altogether, the contributions of this paper are:[1]

1. *An argument search framework.* We present an extensible framework for applying and evaluating research on argument search.
2. *An argument search index.* We provide an index of 291,440 arguments, to our knowledge the largest argument resource available so far.
3. *A prototype argument search engine.* We develop a search engine for arguments, the first that allows retrieving arguments from the web.

2 Related Work

Teufel (1999) was one of the first to point out the importance of argumentation in retrieval contexts, modeling so called argumentative zones of scientific articles. Further pioneer research was conducted by Rahwan et al. (2007), who foresaw a *world wide argument web* with structured argument ontologies and tools for creating and analyzing arguments — the semantic web approach to argumentation. Meanwhile, key parts of the approach have surfaced: the argument interchange format (AIF), a large collection of human-annotated corpora, and tool support, together called *AIFdb* (Bex et al., 2013). Part of AIFdb is a query interface to *browse arguments in the corpora* based on words they contain.[2] In contrast, we face a "real" *search for arguments*, i.e., the retrieval of arguments from the web that fulfill information needs. AIFdb and our framework serve complementary purposes; an integration of the two at some point appears promising.

Web search is the main subject of research in information retrieval, centered around the ranking of web pages that are relevant to a user's information need (Manning et al., 2008). While the scale of the web comes with diverse computational and infrastructural challenges (Brin and Page, 1998), in this paper we restrict our view to the standard architecture needed for the indexing process and the retrieval process of web search (Croft et al., 2009). Unlike standard search engines, though, we index and retrieve arguments, not web pages. The challenges of argument search resemble those IBM's debating technologies address (Rinott et al., 2015). Unlike IBM, we build an open research environment, not a commercial application.

For indexing, a common argument representation is needed. Argumentation theory proposes a number of major models: Toulmin (1958) focuses on fine-grained roles of an argument's units, Walton et al. (2008) capture the inference scheme that an argument uses, and Freeman (2011) investigates how units support or attack other units or arguments. Some computational approaches adopt one of them (Peldszus and Stede, 2015; Habernal and Gurevych, 2015). Others present simpler, application-oriented models that, for instance, distinguish claims and evidence only (Rinott et al., 2015). From an abstract viewpoint, all models share that they consider a single argument as a conclusion (in terms of a claim) together with a set of premises (reasons). Similar to the AIF mentioned above, we thus rely on this basic premise-conclusion model. AIF focuses on inference schemes, whereas we allow for flexible model extensions, as detailed in Section 3. Still, AIF and our model largely remain compatible.

To fully exploit the scale of the web, the arguments to be indexed will have to be mined by a

[1] The framework, index, and search engine can be accessed at: http://www.arguana.com/software.html

[2] AIFdb query interface: http://www.aifdb.org

crawler. A few argument mining approaches deal with online resources. Among these, Boltužić and Šnajder (2014) as well as Park and Cardie (2014) search for supporting information in online discussions, and Swanson et al. (2015) mine arguments on specific issues from such discussions. Habernal and Gurevych (2015) study how well mining works across genres of argumentative web text, and Al-Khatib et al. (2016) use distant supervision to derive training data for mining from a debate portal. No approach, however, seems robust enough, yet, to obtain arguments reliably from the web. Therefore, we decided not to mine at all for our initial index. Instead, we follow the distant supervision idea to obtain arguments automatically.

The data we compile is almost an order of magnitude larger than the aforementioned AIFdb corpus collection currently, and similar in size to the Internet Argument Corpus (Walker et al., 2012). While the latter captures dialogical structure in debates, our data has actual argument structure, making it the biggest argument resource we are aware of.

The core task in the retrieval process is to rank the arguments that are relevant to a query. As surveyed by Wachsmuth et al. (2017a), several quality dimensions can be considered for arguments, from their logical cogency via their rhetorical effectiveness, to their dialectical reasonableness. So far, our prototype search engine makes use of a standard ranking scheme only (Robertson and Zaragoza, 2009), but recent research hints at future extensions: In (Wachsmuth et al., 2017b), we adapt the PageRank method (Page et al., 1999) to derive an objective relevance score for arguments from their relations, ranking arguments on this basis. Boltužić and Šnajder (2015) cluster arguments to find the most prominent ones, and Braunstain et al. (2016) model argumentative properties of texts to better rank posts in community question answering. Others build upon logical frameworks in order to find accepted arguments (Cabrio and Villata, 2012) or credible claims (Samadi et al., 2016).

In addition to such structural approaches, some works target intrinsic properties of arguments. For instance, Feng and Hirst (2011) classify the inference scheme of arguments based on the model of Walton et al. (2008). Persing and Ng (2015) score the argument strength of persuasive essays, and Habernal and Gurevych (2016) predict which of a pair of arguments is more convincing. Such approaches may be important for ranking.

Concept	Description
Argument	
ID	Unique argument ID.
Conclusion	Text span defining the conclusion.
Premises	$k \geq 0$ text spans defining the premises.
Stances	$k \geq 0$ labels, defining each premise's stance.
Argument context	
Discussion	Text of the web page the argument occurs in.
URL	Source URL of the text.
C'Position	Start + end index of the conclusion in the text.
P'Positions	$k \geq 0$ start + end indices, once per premise.
Previous ID	ID of preceding argument in the text if any.
Next ID	ID of subsequent argument in the text if any.
Model extensions (exemplary)	
P'Roles	$k \geq 0$ labels, defining each premise's role.
Scheme	Label defining the argument's scheme.
Scores	$m \geq 0$ values from $[0, 1]$, defining scores.

Table 1: Concepts in our model of an argument and its context as well as examples of model extensions.

3 A Framework for Argument Search

We now introduce the framework that we propose for conducting research related to argument search on the web. It relies on a common argument model and on a standard indexing and retrieval process.

3.1 A Common Argument Model

The basic items to be retrieved by the envisaged kind of search engines are arguments, which hence need to be indexed in a uniform way. We propose a general, yet extensible model to which all arguments can be mapped. The model consists of two parts, overviewed in Table 1, and detailed below.

Argument Each argument has an *ID* and is composed of two kinds of units: a *conclusion* (the argument's claim) and $k \geq 0$ *premises* (reasons). Both the conclusion and the premises may be implicit but not all units. Each premise has a *stance* towards the conclusion (pro or con).[3]

Argument Context We represent an argument's context by the full text of the web page it occurs on (called *discussion* here) along with the page's *URL*.[4] To locate the argument, we model the character indices of conclusions and premises (*C'Position*, *P'Positions*) and we link to the preceding and subsequent argument in the text (*Previous ID*, *Next ID*).

[3] We specify stance only for premises, because a conclusion's stance depends on the issue the argument is used for. For instance, the "right to live" conclusion from Figure 1 supports "abolish death penalty" but it attacks "reintroduce death penalty." For these issues, it takes the role of a premise.

[4] By including the full text, the context of an argument can directly be considered during retrieval. An index, however, would store only a reference to avoid redundancy.

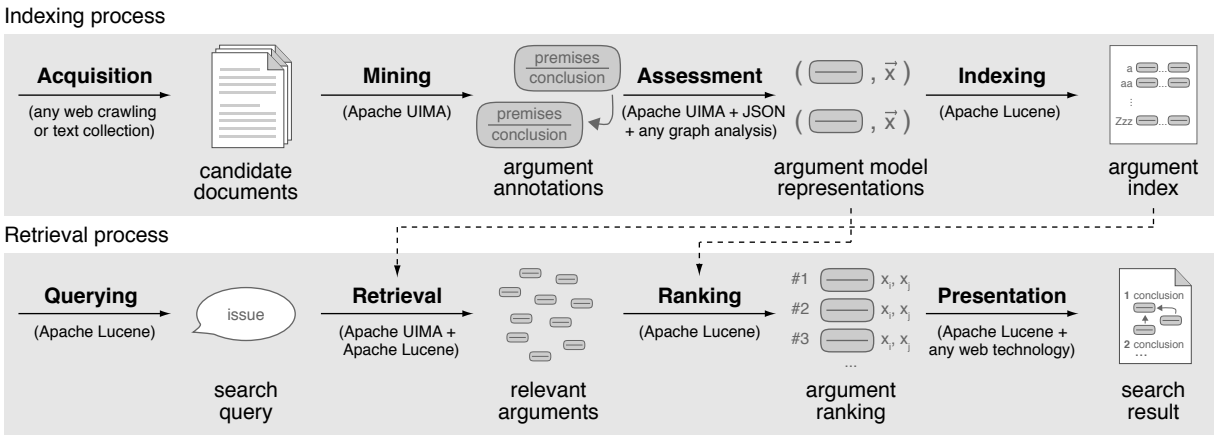

Figure 2: Illustration of the main steps and results of the indexing process and the retrieval process of our argument search engine framework. In parentheses: Technology presupposed in our implementation.

This model represents the common ground of the major existing models (see Section 2), hence abstracting from concepts specific to these models. However, as Table 1 exemplifies, we allow for model extensions to integrate most of them, such as the roles of Toulmin (1958) or the schemes of Walton et al. (2008). Similarly, it is possible to add the various scores that can be computed for an argument, such as different quality ratings (Wachsmuth et al., 2017a). This way, they can still be employed in the assessment and ranking of arguments.

A current limitation of our model pertains to the support or attack between *arguments* (as opposed to *argument units*), investigated by Freeman (2011) among others. While these cannot be represented perfectly in the given model, a solution is to additionally index relations between arguments. We leave such an extension to future work.

3.2 The Indexing Process

Figure 2 concretizes the two standard processes of web search (Croft et al., 2009) for the specific tasks in argument search. The indexing process consists of the *acquisition* of documents, the *mining* and *assessment* of arguments, and the actual *indexing*.

Acquisition The first task is the acquisition of candidate documents, from which the arguments to be indexed are taken. Web search engines employ crawlers to continuously acquire new web pages and to update pages crawled before. The output of this step will usually be HTML-like files or some preprocessed intermediate format. In principle, any text collection in a parsable format may be used.

Mining Having the candidate documents, argument mining is needed to obtain arguments. Several approaches to this task exist as well as to subtasks thereof, such as argument unit segmentation (Ajjour et al., 2017). These approaches require different text analyses as preprocessing. We thus rely on Apache UIMA for this step, which allows for a flexible composition of natural language processing algorithms. UIMA organizes algorithms in a (possibly parallelized) pipeline that iteratively processes each document and adds annotations such as tokens, sentences, or argument units. It is a de facto standard for natural language processing (Ferrucci and Lally, 2004), and it also forms the basis of other text analysis frameworks, such as DKPro (Eckart de Castilho and Gurevych, 2014).

UIMA will allow other researchers to contribute, simply by supplying UIMA implementations of approaches to any subtasks, as long as their output conforms to the set of annotations needed to instantiate our argument model. By collecting implementations for more and more subtasks over time, we aim to build a shared argument mining library.

Assessment State-of-the-art retrieval does not only match web pages with queries, but it also uses meta-properties pre-computed for each page, e.g., the probability of a page being spam, a rating of its reputation, or a query-independent relevance score. For arguments, different structural and intrinsic quality criteria may be assessed, too, as summarized in Section 2. Often, such assessments can be computed from individual arguments, again using UIMA. But some may require an analysis of the graph induced by *all* arguments, such as the PageRank adaptation for arguments we presented (Wachsmuth et al., 2017b). This is why we separate the assessment from the preceding mining step. At

the end, the argument annotations as well as the computed scores are returned in a serializable format (JSON) representing our extended argument model to be fed to the indexer.

Indexing Finally, we create an index of all arguments from their representations, resorting to Apache Lucene due to its wide proliferation. While Lucene automatically indexes all fields of its input (i.e., all concepts of our argument model), the conclusion, the premises, and the discussion will naturally be the most relevant. In this regard, Lucene supplies proven defaults but also allows for a fine-grained adjustment of what is indexed and how.

3.3 The Retrieval Process

The lower part of Figure 2 illustrates the retrieval process of our search framework. When a user *queries* for a controversial issue or similar, relevant arguments are *retrieved*, *ranked*, and *presented*.

Querying We assume any free text query as input. The standard way to process such a query is to interpret it as a set of words or phrases. This is readily supported by Lucene, although some challenges remain, such as how to segment a query correctly into phrases (Hagen et al., 2012). In the context of argument search, the standard way seems perfectly adequate for simple topic queries (e.g., "life-long imprisonment"). However, how people query for arguments exactly and what information needs they have in mind is still largely unexplored. Especially, we expect that many queries will indicate a stance already (e.g., "death penalty is bad" or "abolish death penalty"), ask for a comparison (e.g., "death penalty vs. life-long imprisonment"), or both ("imprisonment better than death penalty").

As a result, queries may need to be preprocessed, for instance, to identify a required stance inversion. Our framework provides interfaces to extend Lucene's query analysis capabilities in this regard. Aside from query interpretation, user profiling may play a role in this step, in order to allow for personalized ranking, but this is left to future work.

Retrieval For a clear separation of concerns, we conceptually decouple argument retrieval from argument ranking. We see the former as the determination of those arguments from the index that are generally relevant to the query. On one hand, this pertains to the problems of term matching known from classic retrieval, including spelling correction, synonym detection, and further (Manning et al., 2008). On the other hand, argument-specific retrieval challenges arise. For instance, what index fields to consider may be influenceed by a query (e.g., "*conclusions* on death penalty"). Our framework uses Lucene for such configurations. Also, we see as part of this step the stance classification of retrieved arguments towards a queried topic (and a possibly given stance), which was in the focus of recent research (Bar-Haim et al., 2017). To analyze arguments, UIMA is employed again.

Ranking The heart of every search engine is its ranker for the retrieved items (here: the arguments). Lucene comes with a number of standard ranking functions for web search and allows for integrating alternative ones. Although a few approaches exist that rank arguments for a given issue or claim (see Section 2), it is still unclear how to determine the most relevant arguments for a given query. Depending on the query and possibly the user, ranking may exploit the content of an argument's conlusion and premises, the argument's context, meta-properties assessed during indexing (see above), or any other metadata. Therefore, this step's input is the full model representations of the retrieved arguments. Its output is a ranking score for each of them.

The provision of a means to apply and evaluate argument ranking functions in practice is one main goal of our framework. An integration of empirical evaluation methods will follow in future work. While we published first benchmark rankings lately (Wachsmuth et al., 2017b), datasets of notable size for this purpose are missing so far.

Presentation Given the argument model representations together with the ranking scores, the last step is to present the arguments to the user along with adequate means of interaction. As exemplified in Figure 1 and 2, both textual and visual presentations may be considered. The underlying snippets of textual representations can be generated with default methods or extensions of Lucene. We do not presuppose any particular web technology for the user interface. Our own approach focusing on the ranking and contrasting of pro and con arguments is detailed in Section 5.

4 An Initial Argument Search Index

The framework from Section 3 serves as a platform for research towards argument search on the web. This section describes an initial data basis that we crawled for carrying out such research. To obtain this data basis, we unified diverse web arguments based on our common argument model.

4.1 Crawling of Online Debate Portals

Being the core task in computational argumentation, argument mining is one of the main analyses meant to be deployed within our framework. As outlined in Section 2, however, current approaches are not yet reliable enough to mine arguments from the web. Following related work (Habernal and Gurevych, 2015; Al-Khatib et al., 2016), we thus automatically derive arguments from the structure given in online debate portals instead.

In particular, we crawled all debates found on five of the largest portals: (1) *idebate.org*, (2) *debatepedia.org*, (3) *debatewise.org*, (4) *debate.org*, and (5) *forandagainst.com*. Except for the second, which was superseded by idebate.org some years ago, these portals have a living community. While the exact concepts differ, all five portals organize pro and con arguments for a given issue on a single debate page. Most covered issues are either of ongoing societal relevance (e.g., "abortion") or of high temporary interest (e.g., "Trump vs. Clinton"). The stance is generally explicit.[5]

The first three portals aim to provide comprehensive overviews of the best arguments for each issue. These arguments are largely well-written, have detailed reasons, and are often supported by references. In contrast, the remaining two portals let users discuss controversies. While on debate.org any two users can participate in a traditional debate, forandagainst.com lets users share own arguments and support or attack those of others.

Although all five portals are moderated to some extent, especially the latter two vary in terms of argument quality. Sometimes users vote rather than argue ("I'm FOR it!"), post insults, or just spam. In addition, not all portals exhibit a consistent structure. For instance, issues on debate.org are partly specified as claims ("Abortion should be legal"), partly as questions ("Should Socialism be preferred to Capitalism?"), and partly as controversial issues ("Womens' rights"). This reflects the web's noisy nature which argument search engines will have to cope with. We therefore index all five portals, taking their characteristics into account.[6]

[5]Other portals were not considered for different reasons. For instance, *createdebate.com* does not represent stance in a pro/con manner, but it names the favored side instead. Hence, an *automatic* conversion into instances of our argument model from Section 3 is not straightforward.

[6]Although not a claim, an issue suffices as a conclusion given that the stance of a premise is known. In contrast, the interpretation of a question as a conclusion may be unclear (e.g., "Why is Confucianism not a better policy?").

4.2 Indexing of Reliable Web Arguments

Given all crawled debates, we analyzed the web page structure of each underlying portal in order to identify how to reliably map the comprised arguments to our common argument model for indexing. An overview of all performed mappings is given in Table 2. For brevity, we only detail the mapping for debatewise.org.[7]

In the majority of debates on debatewise.org, the *debate title* is a claim, such as "Same-sex marriage should be legal". *Yes points* and *no points* are listed that support and attack the claim respectively. For each point, we created one argument where the title is the conclusion and the point is a single premise with either pro stance (for yes points) or con stance (no points). In addition, each point comes with a *yes because* and a *no because*. For a yes point, yes because gives reasons why it holds; for a no point, why it does not hold (in case of no because, vice versa). We created one argument with yes because as the premise and one with no because as the premise, both with the respective point as conclusion. We set the premise stance accordingly.

We abstained from having multiple premises for the arguments derived from any of the portals. Though some reasons are very long and, in fact, often concatenate two or more premises, an automatic segmentation would not be free of errors, which we sought to avoid for the first index. Nevertheless, the premises can still be split once a sufficiently reliable segmentation approach is at hand.

As a result of the mapping, we obtained a set of 376,129 candidate arguments for indexing. To reduce noise that we observed in a manual analysis of samples, we then conducted four cleansing steps: (1) Removal of 368 candidates (from debatepedia.org) whose premise stance could not be mapped automatically to pro or con (e.g., "Clinton" for the issue "Clinton is better than Trump"). (2) Removal of 46,169 candidates whose conclusion is a question, as these do not always constitute proper arguments. (3) Removal of 9930 candidates where either the conclusion or the premise was empty, in order to avoid implicit units in the first index. (4) Removal of 28,222 candidates that were stored multiple times due to the existence of 2852 duplicate debates on debate.org.

Table 3 lists the number of arguments finally indexed from each debate portal, along with the

[7]Besides the actual argument, we also stored all context information reflected in our model, such as the debate's URL.

# Debate Portal	Concept	Mapping to our Common Argument Model	
1 idebate.org	Debate title	Conclusion	of each argument where a pro/con claim is the premise.
	Point for	Pro premise	of one argument where the debate title is the conclusion.
		Conclusion	of the argument where the associated point is the premise.
		Conclusion	of the argument where the associated counterpoint is the premise.
	Point against	Con premise	of one argument where the debate title is the conclusion.
		Conclusion	of the argument where the associated point is the premise.
		Conclusion	of the argument where the associated counterpoint is the premise.
	Point	Pro premise	of the argument where the associated point for/against is the conclusion.
	Counterpoint	Con premise	of the argument where the associated point for/against is the conclusion.
2 debatepedia.org	Debate title	Conclusion	of each argument where a pro/con claim is the premise.
	Pro claim	Pro premise	of one argument where the debate title is the conclusion.
		Conclusion	of the argument where the associated premises are the premise.
	Con claim	Con premise	of one argument where the title is the conclusion.
		Conclusion	of the argument where the associated premises are the premise.
	Premises	Pro premise	of the argument where the associated pro/con claim is the conclusion.
3 debatewise.org	Debate title	Conclusion	of each argument where a pro/con claim is the premise.
	Yes point	Pro premise	of one argument where the debate title is the conclusion.
		Conclusion	of the argument where the associated yes because is the premise.
		Conclusion	of an argument where the associated no because is the premise.
	No point	Con premise	of one argument where the debate title is the conclusion.
		Conclusion	of an argument where the associated yes because is the premise.
		Conclusion	of an argument where the associated no because is the premise.
	Yes because	Pro/Con prem.	of the argument where the associated yes/no point is the conclusion.
	No because	Pro/Con prem.	of the argument where the associated no/yes point is the conclusion.
4 debate.org	Debate title	Conclusion	of each argument of a debate.
	Pro argument	Pro premise	of one argument where the debate title is the conclusion.
	Con argument	Con premise	of one argument where the debate title is the conclusion.
5 forandagainst.com	Claim	Conclusion	of each argument of a debate.
	For	Pro premise	of one argument where the claim is the conclusion.
	Against	Con premise	of one argument where the claim is the conclusion.

Table 2: The concepts given in each debate portal and the mapping we performed to derive arguments.

#	Debate Portal	Argument Units	Arguments	Debates
1	idebate.org	16 084	15 384	698
2	debatepedia.org	34 536	33 684	751
3	debatewise.org	39 576	33 950	2 252
4	debate.org	210 340	182 198	28 045
5	forandagainst.com	29 255	26 224	3 038
$\sum$	Complete index	329 791	291 440	34 784

Table 3: Argument units, arguments, and debates from each portal stored in our initial search index.

number of different argument units composed in the arguments and the number of debates they are taken from. On average, the indexed conclusions and premises have a length of 7.4 and 202.9 words respectively. With a total of 291,440 arguments, to the best of our knowledge, our index forms the largest argument resource available so far.

Naturally, not all indexed arguments have the quality of those from manually annotated corpora. Particularly, we observed that some texts contain phrases specific to the respective debate portal that seemed hard filter out automatically with general rules (e.g., "if we both forfeit every round"). Still, as far as we could assess, the vast majority matches the concept of an argument, which lets our index appear suitable for a first argument search engine.

5 args — The Argument Search Engine

As a proof of concept, we implemented the prototype argument search engine *args* utilizing our framework and the argument index. This section outlines the main features of *args* and reports on some first insights obtained from its usage.[8]

5.1 Content-based Argument Search

The debate portal arguments in our index were collected by a focused crawler and stored directly in the JSON format for indexing. As per our framework, the prototype implements the retrieval process steps of argument search outlined in Section 3 and shown in the lower part of Figure 2.

Querying At server side, our search engine exposes an API, allowing for free text queries to be submitted via HTTP. As on traditional search engines, the entered terms are interpreted as an *AND*

[8] *args* is available at http://www.arguana.com/args. Notice that the prototype is under ongoing development and periodically updated. As a consequence, some of the features described here may change over time.

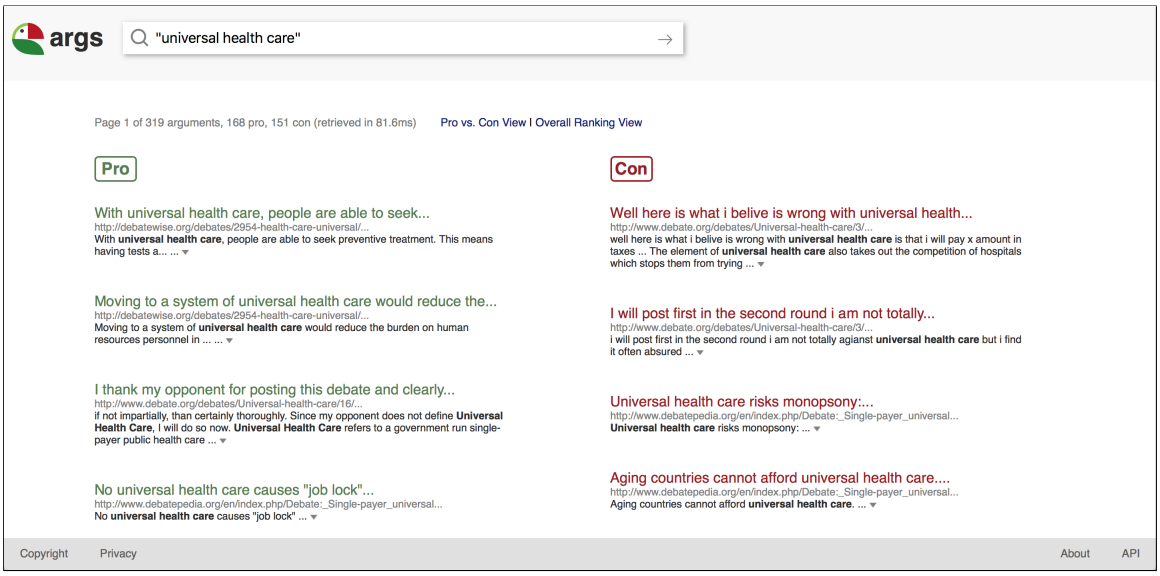

Figure 3: The user interface of the prototype argument search engine *args*, showing the *Pro vs. Con View*.

query, but more search operators are implemented, such as quotes for a *phrase query*. Unlike traditional search engines, stop words are not ignored, since they may be subtle indicators in argumentation (e.g., "arguments *for* feminism").

Retrieval Currently, our prototype retrieves arguments with exact matches of the query terms or phrases. The matching is performed based on conclusions only, making the relevance of the returned arguments to the query very likely. As detailed below, we explored different weightings of the indexed fields though. We derive an argument's stance so far from the stance of its premises stored in our index, which serves as a good heuristic as long as the given query consists of a topic only.

Ranking Before working on rankings based on the specific characteristics of arguments, we seek to assess the benefit and limitations of standard ranking functions for arguments. We rely on *Okapi BM25* here, a sophisticated version of TF-IDF that has proven strong in content-based information retrieval (Croft et al., 2009). In particular, we compute ranking scores for all retrieved arguments with *BM25F*. This variant of BM25 allows a weighting of fields, here of conclusions, premises, and discussions (Robertson and Zaragoza, 2009).

Presentation As a client, we offer the user interface in Figure 3. Right now, search results are presented in two ways: By default, the *Pro vs. Con View* is activated, displaying pro and con arguments separately, opposing each other. In contrast, the *Overall Ranking View* shows an integrated ranking of all arguments, irrespective of stance, making their actual ranks explicit. Views could be chosen automatically depending on the query and user, but this is left to future work. The snippet of a result is created from the argument's premises. A click on the attached arrow reveals the full argument.

5.2 First Insights into Argument Search

Given the prototype, we carried out a quantitative analysis of the arguments it retrieves for controversial issues. The goal was *not* to evaluate the rankings of arguments or their use for downstream applications, since the prototype does not perform an argument-specific ranking yet (see above). Rather, we aimed to assess the coverage of our index and the importance of its different fields. To obtain objective insights we did not compile queries manually nor did we extract them from the debate portals, but referred to an unbiased third party: Wikipedia. In particular, we interpreted all 1082 different controversial issues, which are listed on Wikipedia, as query terms (access date June 2, 2017).[9] Some of these issues are general, such as "nuclear energy" or "drones", others more specific, such as "Park51" or "Zinedine Zidane".

For each issue, we posed a phrase query (e.g., "zinedine zidane"), an AND query (e.g., "zinedine" *and* "zidane"), and an OR query (e.g., "zinedine" *or* "zidane"). Arguments were retrieved using three weightings of BM25F that differ in the fields taken into account: (1) the conclusion field only, (2) the

[9]Issue list: https://en.wikipedia.org/wiki/Wikipedia:List_of_controversial_issues

	Conclusions		Arguments		Contexts	
Query Type	≥ 1	$\tilde{x}$	≥ 1	$\tilde{x}$	≥ 1	$\tilde{x}$
Phrase query	41.6%	24	77.6%	40	77.9%	269
AND query	45.1%	27	88.2%	53	90.0%	498
OR query	84.6%	237	98.0%	1249	98.1%	8800

Table 4: Percentage of the controversial issues on Wikipedia, for which at least one argument is retrieved by our prototype (≥ 1) as well as the median number of arguments retrieved then ($\tilde{x}$); once for each query type based on the *conclusions* only, the full *arguments*, and the full argument *contexts*.

full arguments (i.e., conclusions and premises), and (3) the full contexts (discussions). For all combinations of query type and fields, we computed the proportion of queries, for which arguments were retrieved, and the median number of arguments retrieved then. Table 4 lists the results.

With respect to the different fields, we see that the conclusions, although being short, match with 41.6%–84.6% of all queries, depending on the type of query. Based on the full argument, even phrase queries achieve 77.6%. These numbers indicate that the coverage of our index is already very high for common controversial issues. Moreover, a comparison of the median number of arguments there (40) to those retrieved based on the full context (269) suggests that many other possibly relevant arguments are indexed that do not mention the query terms themselves. While the numbers naturally increase from phrase queries over AND queries to OR queries, our manual inspection confirmed the intuition that especially lower-ranked results of OR queries often lack relevance (which is why our prototype focuses on the other types).

In terms of the weighting of fields, it seems like the highest importance should be given to the conclusion, whereas the discussion should only receive a small weight, but this is up to further evaluation. In general, we observed a tendency towards ranking short arguments higher, implicitly caused by BM25F. Even though, in cases of doubt, short arguments are preferable, we expect that the most relevant arguments need some space to lay out their reasoning. However, to investigate such hypotheses, ranking functions are required that go beyond the words in an argument and its context.

6 Conclusion and Outlook

Few applications exist that exploit the full potential of computational argumentation so far. This paper has introduced a generic argument search framework that is meant to serve as a shared platform for bringing research on computational argumentation to practice. Based on a large index of arguments crawled from the web, we have implemented a prototype search engine to demonstrate the capabilities of our framework. Both the index and the prototype can be freely accessed online.

Currently, however, the index covers only semi-structured arguments from specific debate portals, whereas the prototype is restricted to standard retrieval. While the framework, index, and prototype are under ongoing development, much research on argument mining, argument ranking, and other tasks still has to be done, in order to provide relevant arguments in future search engines.

Laying a solid foundation for research is crucial, since the biggest challenges of argument search transcend basic keyword retrieval. They include advanced retrieval problems, such as learning to rank, user modeling, and search result personalization — all problems with intricate ethical issues attached. Much more than traditional information systems, argument search may affect the convictions of its users. A search engine can be built to do so either blindly, by exposing users to its ranking results as is, or intentionally, by tailoring results to its users. Neither of the two options is harmless:

Training a one-fits-all ranking function on the argumentative portion of the web and on joint user behaviors will inevitably incorporate bias from both the web texts and the dominating user group, affecting the search results seen by the entire user base. On the other hand, tailoring results to individual users would induce a form of confirmation bias: Presuming that the best arguments of either side will be ranked high, should a user with a left-wing predisposition see the left-wing argument on first rank, or the right-wing one? In other words, should a search engine "argue" like the devil's advocate or not? This decision is of utmost importance; it will not only affect how users perceive the quality of the results, but it may also change the stance of the users on the issues they query for. And this, finally, raises the question as to what are actually the *best* arguments: only those that reasonably conclude from acceptable premises — or also those that may be fallacious, yet, persuasive?

Computational argumentation needs to deal with these topics. We believe that this should be done in a collaborative, application-oriented environment.

References

Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein. 2017. Unit Segmentation of Argumentative Texts. In *Proceedings of the Fourth Workshop on Argument Mining*. Association for Computational Linguistics.

Khalid Al-Khatib, Henning Wachsmuth, Matthias Hagen, Jonas Köhler, and Benno Stein. 2016. Cross-Domain Mining of Argumentative Text through Distant Supervision. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 1395–1404. https://doi.org/10.18653/v1/N16-1165.

Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 2017. Stance Classification of Context-Dependent Claims. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, pages 251–261. http://aclweb.org/anthology/E17-1024.

Trevor Bench-Capon, Katie Atkinson, and Peter McBurney. 2009. Altruism and Agents: An Argumentation Based Approach to Designing Agent Decision Mechanisms. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2, AAMAS 2009*. pages 1073–1080. http://dl.acm.org/citation.cfm?id=1558109.1558163.

Floris Bex, John Lawrence, Mark Snaith, and Chris Reed. 2013. Implementing the Argument Web. *Communications of the ACM* 56(10):66–73.

Filip Boltužić and Jan Šnajder. 2014. Back up your Stance: Recognizing Arguments in Online Discussions. In *Proceedings of the First Workshop on Argumentation Mining*. Association for Computational Linguistics, pages 49–58. https://doi.org/10.3115/v1/W14-2107.

Filip Boltužić and Jan Šnajder. 2015. Identifying Prominent Arguments in Online Debates Using Semantic Textual Similarity. In *Proceedings of the 2nd Workshop on Argumentation Mining*. Association for Computational Linguistics, pages 110–115. https://doi.org/10.3115/v1/W15-0514.

Liora Braunstain, Oren Kurland, David Carmel, Idan Szpektor, and Anna Shtok. 2016. Supporting Human Answers for Advice-Seeking Questions in CQA Sites. In *Proceedings of the 38th European Conference on IR Research*. pages 129–141. https://doi.org/10.1007/978-3-319-30671-1_10.

Sergey Brin and Lawrence Page. 1998. The Anatomy of a Large-scale Hypertextual Web Search Engine. In *Proceedings of the Seventh International Conference on World Wide Web*. pages 107–117. http://dl.acm.org/citation.cfm?id=297805.297827.

Elena Cabrio and Serena Villata. 2012. Combining Textual Entailment and Argumentation Theory for Supporting Online Debates Interactions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, pages 208–212. http://aclweb.org/anthology/P12-2041.

Bruce Croft, Donald Metzler, and Trevor Strohman. 2009. *Search Engines: Information Retrieval in Practice*. Addison-Wesley, USA, 1st edition.

Richard Eckart de Castilho and Iryna Gurevych. 2014. A Broad-Coverage Collection of Portable NLP Components for Building Shareable Analysis Pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*. Association for Computational Linguistics and Dublin City University, pages 1–11. http://www.aclweb.org/anthology/W14-5201.

Vanessa Wei Feng and Graeme Hirst. 2011. Classifying Arguments by Scheme. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 987–996. http://aclweb.org/anthology/P11-1099.

David Ferrucci and Adam Lally. 2004. UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Natural Language Engineering* 10(3–4):327–348.

James B. Freeman. 2011. *Argument Structure: Representation and Theory*. Springer.

Ivan Habernal and Iryna Gurevych. 2015. 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*. Association for Computational Linguistics, pages 2127–2137. https://doi.org/10.18653/v1/D15-1255.

Ivan Habernal and Iryna Gurevych. 2016. 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 (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 1589–1599. https://doi.org/10.18653/v1/P16-1150.

Matthias Hagen, Martin Potthast, Anna Beyer, and Benno Stein. 2012. Towards Optimum Query Segmentation: In Doubt Without. In *21st ACM International Conference on Information and Knowledge Management*. pages 1015–1024. https://doi.org/10.1145/2396761.2398398.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120. http://ilpubs.stanford.edu:8090/422/.

Joonsuk Park and Claire Cardie. 2014. Identifying Appropriate Support for Propositions in Online User Comments. In *Proceedings of the First Workshop on Argumentation Mining*. Association for Computational Linguistics, pages 29–38. https://doi.org/10.3115/v1/W14-2105.

Marius Pasca. 2011. Web-based Open-Domain Information Extraction. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. pages 2605–2606. https://doi.org/10.1145/1963192.1963319.

Andreas Peldszus and Manfred Stede. 2015. Joint Prediction in MST-style Discourse Parsing for Argumentation Mining. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 938–948. https://doi.org/10.18653/v1/D15-1110.

Isaac Persing and Vincent Ng. 2015. Modeling Argument Strength in Student Essays. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 543–552. https://doi.org/10.3115/v1/P15-1053.

Iyad Rahwan, Fouad Zablith, and Chris Reed. 2007. Laying the Foundations for a World Wide Argument Web. *Artificial Intelligence* 171(10):897–921.

Ruty Rinott, Lena Dankin, Carlos Alzate Perez, M. Mitesh Khapra, Ehud Aharoni, and Noam Slonim. 2015. 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*. Association for Computational Linguistics, pages 440–450. https://doi.org/10.18653/v1/D15-1050.

Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval* 3(4):333–389.

Mehdi Samadi, Partha Pratim Talukdar, Manuela M. Veloso, and Manuel Blum. 2016. ClaimEval: Integrated and Flexible Framework for Claim Evaluation Using Credibility of Sources. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. pages 222–228.

Reid Swanson, Brian Ecker, and Marilyn Walker. 2015. Argument Mining: Extracting Arguments from Online Dialogue. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, pages 217–226. https://doi.org/10.18653/v1/W15-4631.

Simone Teufel. 1999. *Argumentative Zoning: Information Extraction from Scientific Text*. Ph.D. thesis, University of Edinburgh.

Stephen E. Toulmin. 1958. *The Uses of Argument*. Cambridge University Press.

Henning Wachsmuth, Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Alberdingk Tim Thijm, Graeme Hirst, and Benno Stein. 2017a. Computational Argumentation Quality Assessment in Natural Language. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, pages 176–187. http://aclweb.org/anthology/E17-1017.

Henning Wachsmuth, Benno Stein, and Yamen Ajjour. 2017b. "PageRank" for Argument Relevance. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, pages 1117–1127. http://aclweb.org/anthology/E17-1105.

Marilyn Walker, Jean Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A Corpus for Research on Deliberation and Debate. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*. European Language Resources Association (ELRA), pages 812–817. http://www.lrec-conf.org/proceedings/lrec2012/pdf/1078_Paper.pdf.

Douglas Walton, Christopher Reed, and Fabrizio Macagno. 2008. *Argumentation Schemes*. Cambridge University Press.

Lu Wang and Wang Ling. 2016. Neural Network-Based Abstract Generation for Opinions and Arguments. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 47–57. https://doi.org/10.18653/v1/N16-1007.