ArgumenText: Searching for Arguments in Heterogeneous Sources

Christian Stab and **Johannes Daxenberger** and **Chris Stahlhut** and **Tristan Miller**
Benjamin Schiller and **Christopher Tauchmann** and **Steffen Eger** and **Iryna Gurevych**
Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technische Universität Darmstadt
https://www.ukp.tu-darmstadt.de/

Abstract

Argument mining is a core technology for enabling argument search in large corpora. However, most current approaches fall short when applied to heterogeneous texts. In this paper, we present an argument retrieval system capable of retrieving sentential arguments for any given controversial topic. By analyzing the highest-ranked results extracted from Web sources, we found that our system covers 89% of arguments found in expert-curated lists of arguments from an online debate portal, and also identifies additional valid arguments.

1 Introduction

Information retrieval (IR) and question answering (QA) are mature NLP technologies that excel at finding factual information relevant to a given query. But not all information needs can be satisfied with factual information. In many search scenarios, users are not seeking a universally accepted ground truth, but rather an overview of viewpoints and arguments surrounding a controversial topic. For example, in a legal dispute, an attorney might have to search for precedents and multifaceted legal opinions supporting the case at hand, and anticipate counterarguments that opposing counsel will make. Similarly, a policymaker will survey pros and cons of prospective legislation before she proposes or votes on it. While IR and QA can help with such *argument search* tasks, they provide no specialized support for them.

Despite its obvious applications, argument search has attracted relatively little attention in the argument mining community. In this paper, we present ArgumenText, which we believe is the first system for topic-relevant argument search in heterogeneous texts. It takes a large collection of arbitrary Web texts, automatically identifies arguments relevant to a given topic, classifies them as “pro” or “con”, and presents them ranked by relevance in an intuitive interface. The system thereby eases much of the manual effort involved in argument search.

We present an evaluation of our system in which its top-ranked search results are compared with arguments aggregated and curated by experts on a popular online debate portal. The results show that our system has high coverage (89%) with respect to the expert-curated lists. Moreover, it identifies many additional valid arguments omitted or overlooked by the human curators, affording users a more complete overview of the controversy surrounding a given topic. Nonetheless, precision remains an issue, with slightly less than half (47%) the results being irrelevant to the topic or misclassified with respect to argument stance.

2 Related Work

Most existing approaches consider argument mining at the discourse level and address tasks like argument unit identification (Ajjour et al., 2017), component classification (Mochales-Palau and Moens, 2009), or argument structure identification (Eger et al., 2017). These approaches focus on recognizing arguments within a single text but do not consider relevance to user-defined topics.

Until now, there has been little work on identifying topic-relevant arguments. Wachsmuth et al. (2017) present a generic framework for argument search that relies on pre-structured arguments from debate portals. Levy et al. (2014) present a system designed specifically for detecting topic-relevant claims from Wikipedia, which was later extended to mine supporting statements for claims (Rinott et al., 2015). The MARGOT system (Lippi and Torroni, 2015) is trained on Wikipedia data and extracts claims and evidence from user-provided texts. However, all these systems focus on specific text types and are not yet able to extract arguments

Proceedings of NAACL-HLT 2018: Demonstrations, pages 21–25
New Orleans, Louisiana, June 2 - 4, 2018. ©2018 Association for Computational Linguistics

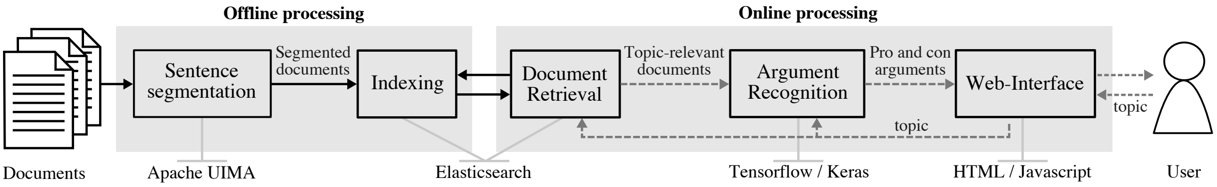


Figure 1: System architecture.

from a large collection of arbitrary texts. The approach most similar to ours, introduced by Hua and Wang (2017), extracts claim-relevant arguments from different text types, but is limited to sentential "pro" arguments.

3 System Description

Our system allows searching for arguments relevant to a user-defined topic. A *topic* is some matter of controversy that can be concisely expressed through keywords. We define an *argument* as a sentence expressing evidence or reasoning that can be used to either *support* or *oppose* a given topic. For example, "It carries a risk of genetic defects." is a "con" argument for the topic "cloning" while "Cloning should be permitted." is not an argument at all since it lacks a relevant reason.

Retrieving arguments from a large document collection is computationally expensive. In particular, argument mining methods that consider the relevance to a specific topic need to be applied for each query individually, resulting in poor response times if the collection is too big. To address this challenge, our system first retrieves a list of documents relevant to a given topic and then applies an argument mining model to the top-ranked documents. The system's architecture (Fig. 1) is split into offline and online processing parts. The *offline processing* consists of components not depending on the user's query such as boilerplate removal, sentence segmentation, and document indexing. The *online processing* covers all components that depend on the user-defined topic and thus need to be applied for each query. The following subsections describe each of these components in detail.

3.1 Data

As our objective is to search for arguments in any text domain, we build upon the English part of CommonCrawl,[1] the largest Web corpus available to date. Before further processing, we followed Habernal et al. (2016) for de-duplication, boilerplate removal using jusText (Pomikálek, 2011), and language detection.[2] This left us with 400 million heterogeneous plain-text documents in English, with an overall size of 683 GiB.

3.2 Tokenization and Sentence Segmentation

Each document is segmented into sentences with an Apache UIMA pipeline using components from DKPro Core (Eckart de Castilho and Gurevych, 2014). To facilitate processing of other languages in future work, we chose Apache OpenNLP which currently supports six languages. The modular nature of our setup allows us to easily integrate other sentence segmentation methods for currently unsupported languages. Finally, the document text, the tokenized sentences, and the metadata (e.g., document titles and timestamps) are converted into a JSON format for indexing.

3.3 Indexing and Retrieval

To retrieve documents relevant to a given topic, we index the data using Elasticsearch.[3] The entire offline processing of our data, using 40 parallel processes on a server equipped with two Intel Xeon E5-2699 v4 CPUs (22 cores each) and 512 GiB of memory, required 19 days in total.

For each request, Elasticsearch scores all documents containing the keywords of the topic according to BM25 (Robertson et al., 1994). It then returns the top-ranked documents, including the segmented sentences and metadata, in the aforementioned JSON format. We can optionally restrict the search to specific fields in the metadata, such as the publication date or source domain.

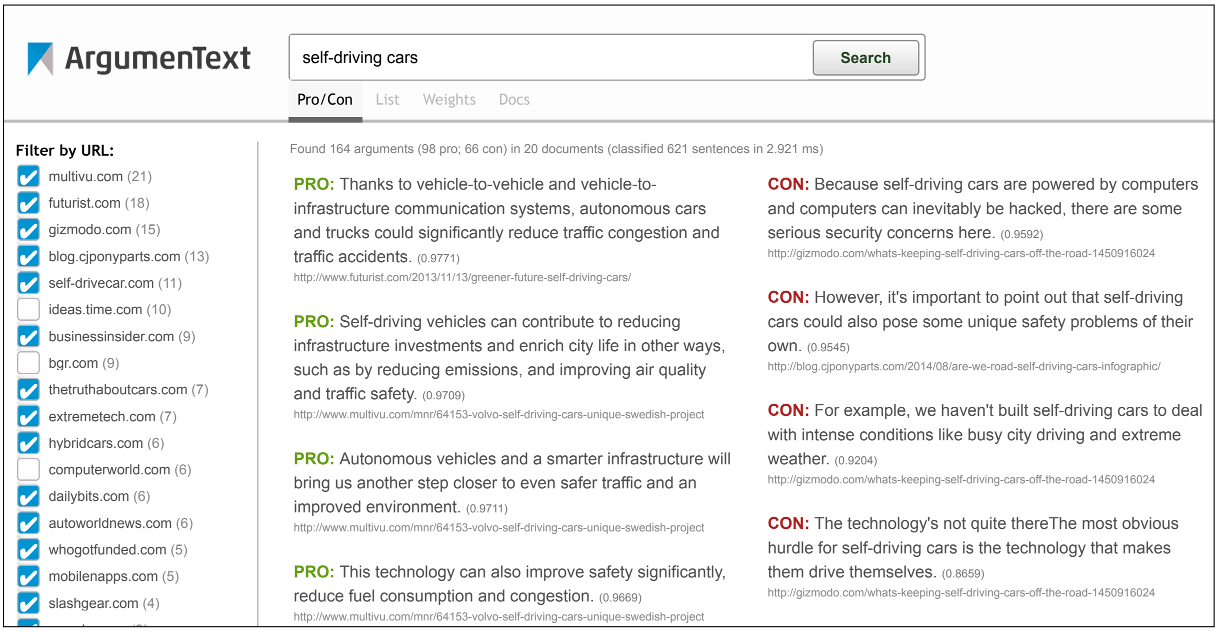
3.4 Argument Identification and Stance Recognition

For extracting topic-relevant arguments from the list of retrieved documents, we build on the corpus of Stab et al. (2018), which includes annotated

[1] http://commoncrawl.org/

[2] We use the Language Detection Library available at https://github.com/shuyo/language-detection.

[3] https://www.elastic.co/



Filter by URL:

☑ multivu.com (21)
☑ futurist.com (18)
☑ gizmodo.com (15)
☑ blog.cjponyparts.com (13)
☑ self-drivecar.com (11)
☐ ideas.time.com (10)
☑ businessinsider.com (9)
☐ bgr.com (9)
☑ thetruthaboutcars.com (7)
☑ extremetech.com (7)
☑ hybridcars.com (6)
☐ computerworld.com (6)
☑ dailybits.com (6)
☑ autoworldnews.com (6)
☑ whogotfunded.com (5)
☑ mobilenapps.com (5)
☑ slashgear.com (4)

Found 164 arguments (98 pro; 66 con) in 20 documents (classified 621 sentences in 2.921 ms)

PRO: Thanks to vehicle-to-vehicle and vehicle-to-infrastructure communication systems, autonomous cars and trucks could significantly reduce traffic congestion and traffic accidents. (0.9771)
http://www.futurist.com/2013/11/13/greener-future-self-driving-cars/

PRO: Self-driving vehicles can contribute to reducing infrastructure investments and enrich city life in other ways, such as by reducing emissions, and improving air quality and traffic safety. (0.9709)
http://www.multivu.com/mnr/64153-volvo-self-driving-cars-unique-swedish-project

PRO: Autonomous vehicles and a smarter infrastructure will bring us another step closer to even safer traffic and an improved environment. (0.9711)
http://www.multivu.com/mnr/64153-volvo-self-driving-cars-unique-swedish-project

PRO: This technology can also improve safety significantly, reduce fuel consumption and congestion. (0.9669)
http://www.multivu.com/mnr/64153-volvo-self-driving-cars-unique-swedish-project

CON: Because self-driving cars are powered by computers and computers can inevitably be hacked, there are some serious security concerns here. (0.9592)
http://gizmodo.com/whats-keeping-self-driving-cars-off-the-road-1450916024

CON: However, it's important to point out that self-driving cars could also pose some unique safety problems of their own. (0.9545)
http://blog.cjponyparts.com/2014/08/are-we-road-self-driving-cars-infographic/

CON: For example, we haven't built self-driving cars to deal with intense conditions like busy city driving and extreme weather. (0.9204)
http://gizmodo.com/whats-keeping-self-driving-cars-off-the-road-1450916024

CON: The technology's not quite thereThe most obvious hurdle for self-driving cars is the technology that makes them drive themselves. (0.8659)
http://gizmodo.com/whats-keeping-self-driving-cars-off-the-road-1450916024

Figure 2: The UI's Pro/Con view, showing "pro" and "con" arguments for the query "self-driving cars".

sentences for eight topics. To cover a wider range of topics, we extended the corpus with 41 additional topics, such as "self-driving cars" and "basic income", using the same procedure: we queried Google for each topic, extracted 600 sentences for each topic from the search results, and had seven crowd workers annotate each sentence as either a "pro" argument, a "con" argument, or not an argument. As in Stab et al. (2018), we used MACE (Hovy et al., 2013) with a threshold of 0.9 to merge the annotations. This process provided us with an additional 22,691 annotated sentences, of which 27% are annotated as "pro" arguments, 18% as "con" arguments, and 55% as not an argument.

Using this extended corpus, we first trained the attention-based neural network presented by Stab et al. (2018) which classifies each sentence as *argument* or *no argument* with respect to the user-defined topic. Second, we apply a BiLSTM model to determine the stance (*pro* or *con*) of each topic-relevant argument.[4] To evaluate these models, we conduct a leave-one-topic-out evaluation—i.e., we trained the models on $n - 1$ topics and evaluated their performance on the left-out topic. The results show that the models benefit from the broader range of topics in our extended corpus. In particular, the performance of argument identification improves to 73.84 macro F-score as compared to 65.8 macro F-score when trained on the initial corpus with eight topics. The stance model is trained on the "pro" and "con" arguments and achieves an average macro F-score of 76.61 across all topics. It outperforms by a large margin a logistic regression baseline with unigram features achieving 67.92 macro F-score.

3.5 User Interface

The user interface resembles a typical search engine and allows queries for any controversial topic. To provide the user with arguments of the highest confidence, the retrieved arguments are sorted by the average confidence score of the argument extraction and stance recognition model. The user can choose between three argument-based views (1–3) and a document-based view (4):

(1) *Pro/Con view.* This view (Fig. 2) presents the user with a ranked list of "pro" and "con" arguments next to each other. To provide access to the origin and context of arguments, the document URL is displayed for each argument as well as the average confidence score.

(2) *List view.* This view provides the same information as the Pro/Con view, but shows all arguments interleaved in a single list instead of as two separate lists.

(3) *Attention Weights view.* To show which words most influence the classifier in its decision, we visualize attention weights for each word of an argumentative sentence. Important words are underlined in the view; the more intense the colour of the underlining, the more important

[4] Using two different models gave us slightly better results than using a single three-label model.

Topic	# pro	# con
cellphones	75	102
social networking	224	64
animal testing	455	609

Table 1: Arguments considered in the evaluation study.

the word is to the topic. The view is otherwise structured like the Pro/Con view.

(4) *Documents view.* This view ranks documents by the number of arguments they contain. It shows the number of "pro" and "con" arguments in bar charts next to the document titles, which can be expanded to list their arguments.

Each view features a filtering function for excluding arguments from specific sources (e.g., websites the user considers unreliable—see left side of Fig. 2). By default, arguments from all sources are shown.

4 Evaluation

As we believe that our system will be beneficial for a broad range of applications, we decided not to focus on a particular use case for the evaluation. Rather, we compared the output of the system against expert-created argument summaries from the online debate platform ProCon.org. For three randomly selected topics excluded from our training data, we extracted 1529 arguments from our system output (see Table 1). For the same topics, we then collected all expert-created "pro" and "con" arguments from ProCon.org.

In a manual evaluation study with three undergraduate and graduate students of computer science, we assessed the perceived quality of the system-discovered arguments and their overlap with expert-created arguments from ProCon.org.[5] Each student went through the entire list of system-discovered arguments and decided whether each one (i) could be mapped to one or more of the expert-created "pro" arguments; (ii) could be mapped to one or more of the expert-created "con" arguments; (iii) was not an argument, was nonsensical, or had the wrong stance; or (iv) was a completely new argument. Since our interest is in the perceived usefulness of the system rather than its ability to precisely match a carefully crafted gold standard, we simply aggregated votes for each of the above categories and averaged them for the three participants of the study. The results provided some high-level insights about the potential and limitations of the system.

First, we discovered that our system's coverage (i.e., the percentage of expert-created arguments mapped to one or more arguments from our system) is very high—89% across the three topics. "Social networking", with 46 unique expert-curated arguments, was the only topic with less than perfect argument coverage (78%). Second, 12% of the aggregated votes indicated that a sentence is a completely new argument (i.e., a valid argument, not necessarily unique, with no expert-created counterpart)—a strong indicator that our system is not just usable to detect arguments for a broader range of topics as compared to expert-curated platforms, but also to get a more complete picture about individual topics. Third, we also discovered that on average 47% of arguments fell into category (iii), meaning that while the coverage of our system is high, precision is still a problem.

We also assessed the ranking by repeating the evaluation for only the top 10, 50, and 100 arguments. The percentage of system-discovered new arguments is identical across ranks. As for coverage, a bit more than 40% of expert-curated arguments can be found among the first ten results on average, while 71% can be found among the first 50 and 79% among the first 100. While nonsensical sentences were more common at lower ranks, the percentage of both non-arguments and arguments with incorrect stance remains stable across ranks at about 30% and 12%, respectively.

5 Conclusion and Future Work

We have presented ArgumenText,[6] an argument search system capable of retrieving "pro" and "con" arguments relevant to a given topic from heterogeneous sources.[7] By comparing the top-ranked results to arguments from debate portals, we have shown that our system achieves a high coverage compared to expert-created lists of arguments, and that it is even capable of finding additional valid arguments. In future work, we aim to improve the precision of the system by employing more sophisticated deep learning architectures like adversarial neural networks, to experiment with other argument ranking methods, and to adapt the approach to other languages such as German.

[5] For the sake of comparison, we considered only the ProCon.org summary sentence of each argument—e.g., "Animal testing is cruel and inhumane."

[6] Available at http://www.argumentsearch.com

[7] This work has been supported by the German Federal Ministry of Education and Research (BMBF) under the promotional reference 03VP02540 (ArgumenText).

References

Yamen Ajjour, Wei-Fan Chen, Johannes Kiesel, Henning Wachsmuth, and Benno Stein. 2017. Unit segmentation of argumentative texts. In *Proceedings of the 4th Workshop on Argument Mining*. Association for Computational Linguistics, pages 118–128. http://www.aclweb.org/anthology/W17-5115.

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.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural end-to-end learning for computational argumentation mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 11–22. http://aclweb.org/anthology/P17-1002.

Ivan Habernal, Omnia Zayed, and Iryna Gurevych. 2016. C4Corpus: Multilingual Web-size corpus with free license. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*. European Language Resources Association (ELRA), pages 914–922. http://www.lrec-conf.org/proceedings/lrec2016/pdf/388_Paper.pdf.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 1120–1130. http://www.aclweb.org/anthology/N13-1132.

Xinyu Hua and Lu Wang. 2017. Understanding and detecting supporting arguments of diverse types. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, pages 203–208. http://aclweb.org/anthology/P17-2032.

Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. 2014. Context dependent claim detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, pages 1489–1500. http://www.aclweb.org/anthology/C14-1141.

Marco Lippi and Paolo Torroni. 2015. MARGOT: A web server for argumentation mining. *Expert Systems with Applications* 65:292–303. https://doi.org/10.1016/j.eswa.2016.08.050.

Raquel Mochales-Palau and Marie-Francine Moens. 2009. Argumentation mining: The detection, classification and structure of arguments in text. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*. Association for Computing Machinery, pages 98–107. https://doi.org/10.1145/1568234.1568246.

Jan Pomikálek. 2011. *Removing Boilerplate and Duplicate Content from Web Corpora*. Doctoral thesis, Masaryk University, Faculty of Informatics, Brno, Czech Republic. https://is.muni.cz/th/45523/fi_d/phdthesis.pdf.

Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M. 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. http://aclweb.org/anthology/D15-1050.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline M. Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference*. NIST, pages 109–126. http://trec.nist.gov/pubs/trec3/papers/city.ps.gz.

Christian Stab, Tristan Miller, and Iryna Gurevych. 2018. Cross-topic argument mining from heterogeneous sources using attention-based neural networks. *arXiv preprint* 1802.05758. https://arxiv.org/abs/1802.05758.

Henning Wachsmuth, Martin Potthast, Khalid Al Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. 2017. Building an argument search engine for the Web. In *Proceedings of the 4th Workshop on Argument Mining*. Association for Computational Linguistics, pages 49–59. http://www.aclweb.org/anthology/W17-5106.