PERSPECTROSCOPE: A Window to the World of Diverse Perspectives

Sihao Chen, Daniel Khashabi, Chris Callison-Burch, Dan Roth
University of Pennsylvania
{sihaoc,danielkh,ccb,danroth}@cis.upenn.edu

Abstract

This work presents PERSPECTROSCOPE, a web-based system which lets users query a discussion-worthy natural language claim, and extract and visualize various perspectives in support or against the claim, along with evidence supporting each perspective. The system thus lets users explore various perspectives that could touch upon aspects of the issue at hand. The system is built as a combination of retrieval engines and learned textual-entailment-like classifiers built using a few recent developments in natural language understanding. To make the system more adaptive, expand its coverage, and improve its decisions over time, our platform employs various mechanisms to get corrections from the users.

PERSPECTROSCOPE is available at github.com/CogComp/perspectroscope.[1]

1 Introduction

One key consequence of the information revolution is a significant increase and a contamination of our information supply. The practice of fact-checking won't suffice to eliminate the biases in text data we observe, as the degree of factuality alone does not determine whether biases exist in the spectrum of opinions visible to us. To better understand controversial issues, one needs to view them from a diverse yet comprehensive set of *perspectives*.

Understanding most nontrivial *claim*s requires insights from various *perspective*s. Today, we make use of search engines or recommendation systems to retrieve information relevant to a claim, but this process carries multiple forms of *bias*. In particular, they are optimized relative to the claim (query) presented, and the popularity of the relevant documents returned, rather than with respect to the diversity of the *perspective*s presented in them or whether they are supported by evidence.

While it might be impractical to show an exhaustive spectrum of views with respect to a *claim*, cherry-picking a small but diverse set of *perspectives* could be a tangible step towards addressing the limitations of the current systems. Inherently this objective requires the understanding of the relations between each *perspective* and *claim*, as well as the nuance in semantic meaning between *perspective*s under the context of the *claim*.

This work presents a demo for the task of *substantiated perspective discovery* (Chen et al., 2019). Our system receives a *claim* and it is expected to present a *diverse* set of *well-corroborated perspectives* that take a *stance* with respect to the claim. Each perspective should be substantiated by *evidence* paragraphs which summarize pertinent results and facts.

A typical output of the system is shown in Figure 3. The input to the system is a *claim*: *Social media (like facebook or twitter) have had very positive effects in our life style.* There is no single, best way to respond to the claim, but rather there are many valid responses that form a spectrum of perspectives, each with a *stance* relative to this claim and, ideally, with evidence supporting it.

To support the input claim, one could refer to the observation that interactions between individuals has become easier through the social media. Or one can refer to the success they have brought to those in need of reaching out to masses (e.g., business individuals). On the contrary, one could oppose the given claim by pointing out its negative impacts on productivity and the increase in cyber-bullying. Each of these arguments, which we refer to as a *perspective* throughout the paper, is an opinion, possibly conditional, in support of a given *claim* or against it. A *perspective* thus con-

[1] A brief demo of the system: https://www.youtube.com/watch?v=MXBTR1Sp3Bs.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 129–134
Florence, Italy, July 28 - August 2, 2019.

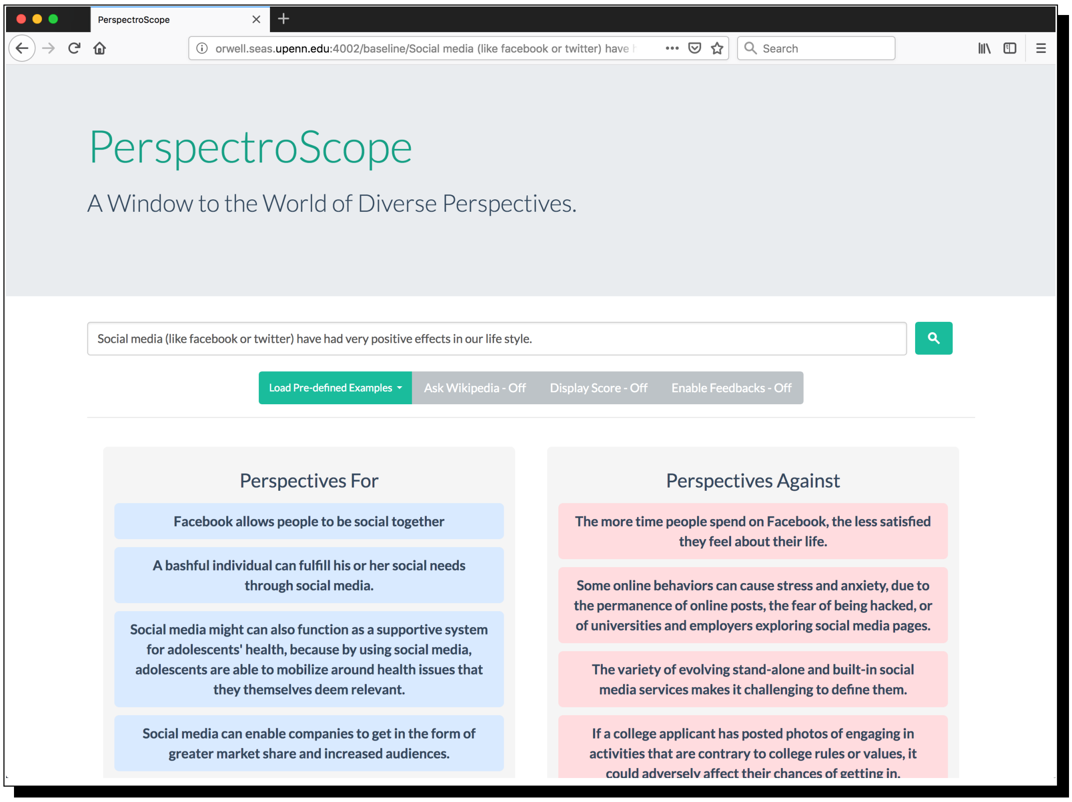

Figure 1: Given a *claim* as the input, our system is expected to discover various *perspectives* and their *stance* with respect to the claim. Each claim also comes with the relevant *evidence* that substantiates the given perspective.

stitutes a particular attitude towards a given *claim*. Additionally, each of these *perspective* has to be well-supported by *evidence* found in paragraphs that summarize findings and substantiations of different sources.

Overall, PERSPECTROSCOPE provides an interface to help individuals by providing a small but diverse set of *perspectives*. Our system is built upon a few recent developments in the field. In addition, our system is designed to be able to utilize feedback from the users of the system to improve its predictions. The rest of this paper is dedicated to delineating the details of PERSPECTROSCOPE.

2 PERSPECTROSCOPE

2.1 Core Design Structure

A high-level picture of the work is shown in Figure 2. Our system uses a mix of retrieval engines and learned classifiers to ensure both quality and efficiency. The retrieval systems extract candidates (perspectives or evidence paragraphs) which are later evaluated by carefully designed classifiers.

2.2 Learned Classifiers

In building PERSPECTROSCOPE we borrow the definitions and dataset provided by Chen et al. (2019). The provided dataset, PERSPECTRUM, is a crowdsourced collection of claims, perspectives and evidence extracted from online debate websites as well as other web content. We follow the same steps as Chen et al. (2019) to create classifiers for the following tasks:

C1: Relevant Perspective Extraction. This classifier is expected to return the collection of perspectives with respect to a given claim.

C2: Perspective Stance Classification. Given a claim, this classifier is expected to score a collection of perspectives with the degree to which it *supports* or *opposes* the given claim.

C3: Perspective Equivalence. This classifier is expected to decide whether two given perspectives are equivalent or not, in the context a given claim.

C4: Extraction of Supporting Evidence. This classifier decides whether a given document lends enough evidence for a given perspective to a claim.

In training the classifiers for each of the tasks,

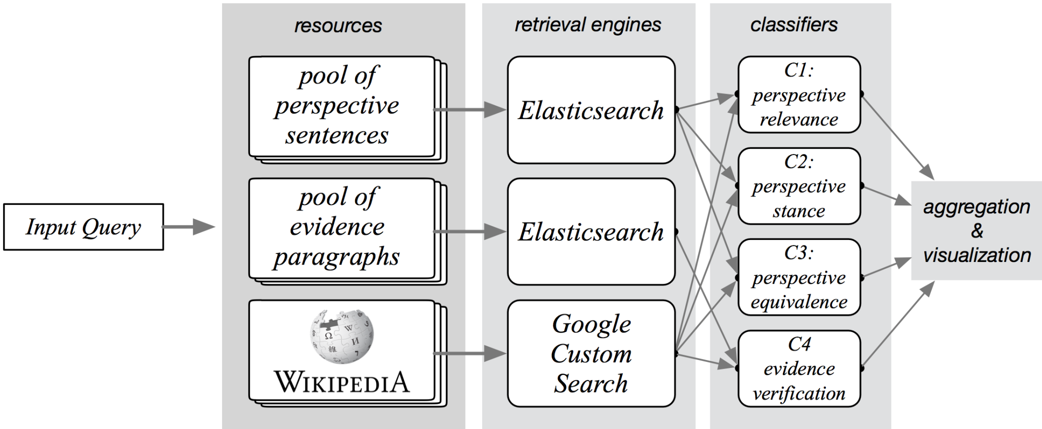

Figure 2: Overview of the system structure: given a query to the system, it extracts candidates from its internal knowledge

we use BERT (Devlin et al., 2019) and we follow the same steps described in Chen et al. (2019).

2.3 Candidate Retrieval

We use a retrieval (IR) system[2] to generate *perspective* and *evidence* candidates for the learned classifiers. We take $10k$ perspective sentences and $8k$ evidence paragraphs from Chen et al. (2019) and index them respectively in two independent retrieval engines. For each input claim, we query the claim and retrieve top-30 perspective candidates from the retrieval engine. Upon user request, we query the claim concatenated with a perspective candidate to retrieve top-20 evidence candidates from the pool of $8k$ evidence paragraphs.

To support a broader range of topics not covered by PERSPECTRUM, we use Wikipedia to retrieve extra candidate perspectives/evidence. Given an input claim from the user, we issue a query to the Google Custom Search API [3] and retrieve top 10 relevant Wikipedia pages. We clean up each page using newspaper3k[4] and use the first sentence of the paragraphs within each document as candidate perspectives, and the rest sentences in each paragraph as candidate evidence.

2.4 Minimal Perspective Discovery

The overall decision making is outlined in Algorithm 1. As mentioned earlier, the whole process is a pipeline starting with candidate generation via retrieval engines, and followed by scoring with the learned classifiers. The final step is to select a *minimal* set of perspectives with the DBSCAN clustering algorithm (Ester et al., 1996).

Algorithm 1: Minimal Perspective Extraction

Input: claim c.
Output: perspectives, their stances & evidence.
$\hat{P} \leftarrow$IR(c) `// candidate perspectives`
$P = \{\}$
foreach $p \in \hat{P}$ **do**
  `// perspective relevance`
  if $C1(c,p) > T1$ *and* $abs(C2(c,p)) > T2$ **then**
    $e \leftarrow C2(c,p)$
    $\hat{E} \leftarrow$IR(c,p) `// candidate evidence`
    $E = \{\}$
    foreach $e \in \hat{E}$ **do**
      `// evidence verification`
      if $C4(c,p,e) > T4$ **then**
        $E \leftarrow E \cup \{e\}$.
      end
    end
    $P \leftarrow P \cup \{(p,s,E)\}$.
  end
end
$P \leftarrow$ `/* minimal perspectives after clustering with DBSCAN on the equivalence scores between any perspective pairs via` $C3$. `*/`

The parameters of this algorithm (e.g., the thresholds $T1, T2, ...$) are tuned manually on a held-out set.

2.5 Utilizing user feedback

User feedback/logs are valuable sources of information for many successful applications. In this work, we collect two forms of feedback signals from users. We record all queries of claims issued to the system. In addition, the users have the option to tell us whether a given perspective is a good or bad one (based on the quality of its relevance, stance or evidence prediction). It is important to

[2] www.elastic.co
[3] https://cse.google.com/cse/
[4] github.com/codelucas/newspaper

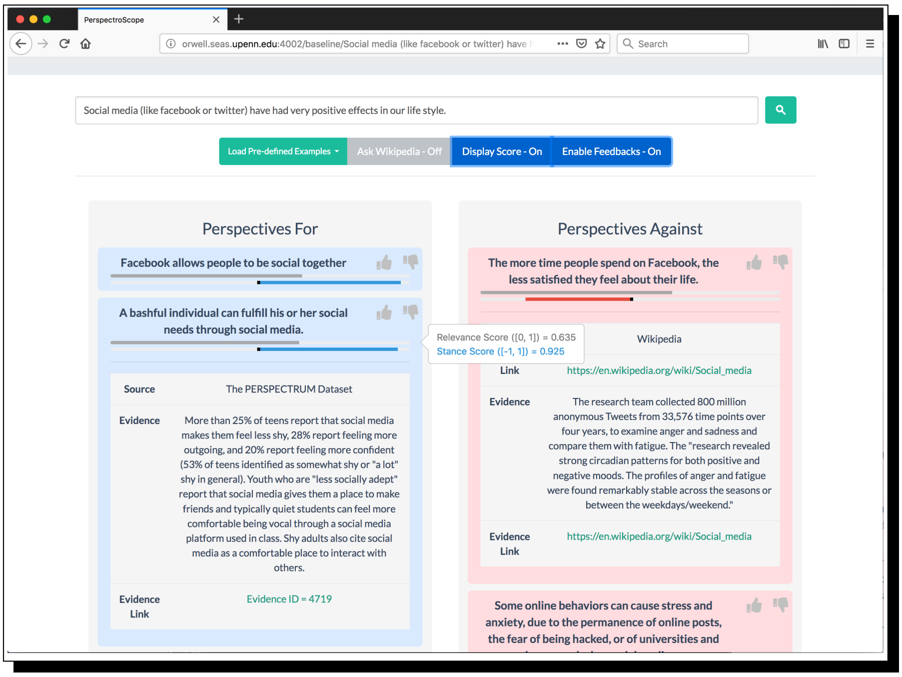

Figure 3: A demonstration of the system features. The grey and blue/red color bars (under each perspective) show the relevance and stance predictions, respectively. Upon user request, the system provides a paragraph of supporting evidence for each perspective. Users have the option to provide feedback to each perspective via the *thumbs-up* or *thumbs-down* button.

note that we are not collecting any personal information in the process.

The user annotations can provide extra supervision signals for task C1-C4 with a broader topical coverage. These annotations can in turn be used in the classifier training and iteratively improve our prediction results with increasing number of users.

3 Related Work

There are few related tools to this work. `args.me` is a platform that accepts natural language queries and returns links to the pages that contain relevant topics (Wachsmuth et al., 2017), which are split into *supporting* & *opposing* categories (screenshot in Figure 4). Similarly, `ArgumentText` (Stab et al., 2018a) takes a topic as input and returns *pro/con* arguments retrieved from the web. This work takes the effort one step further by employing language understanding techniques.

There is a rich line of work on using Wikipedia as source for argument mining or to assess the veracity of a claim (Thorne et al., 2018). For instance, `FAKTA` is a system that extracts relevant documents from Wikipedia, among other sources, to predict the factuality of an input claim (Nadeem et al., 2019).

Beyond published works, there are websites that employ similar technologies. For instance, `bing.com` has recently started a service that provides two different responses to a given argument (screenshot in Figure 4). Since there is no published work on this system, it is not clear what the underlying mechanism is.

There exist a number of online debate platforms that provide similar functionalities as our system: `kialo.com`, `procon.org`, `idebate.org` , among others. Such websites usually provide a wide range of debate topics and various arguments in response to each topic. These resources have been proven useful in a line of works in argumentation (Hua and Wang, 2017; Stab et al., 2018b; Wachsmuth et al., 2018), among many others. While they provide rich sources of information, their content is fairly limited in terms of either their topical coverage or data availability for academic research purposes.

There also exist a few other works in this direction that do not accompany a publicly available tool or demo. For instance, Hasan and Ng (2014); Levy et al. (2018) attempt to identify relevant arguments within web text in the context of a given topic.

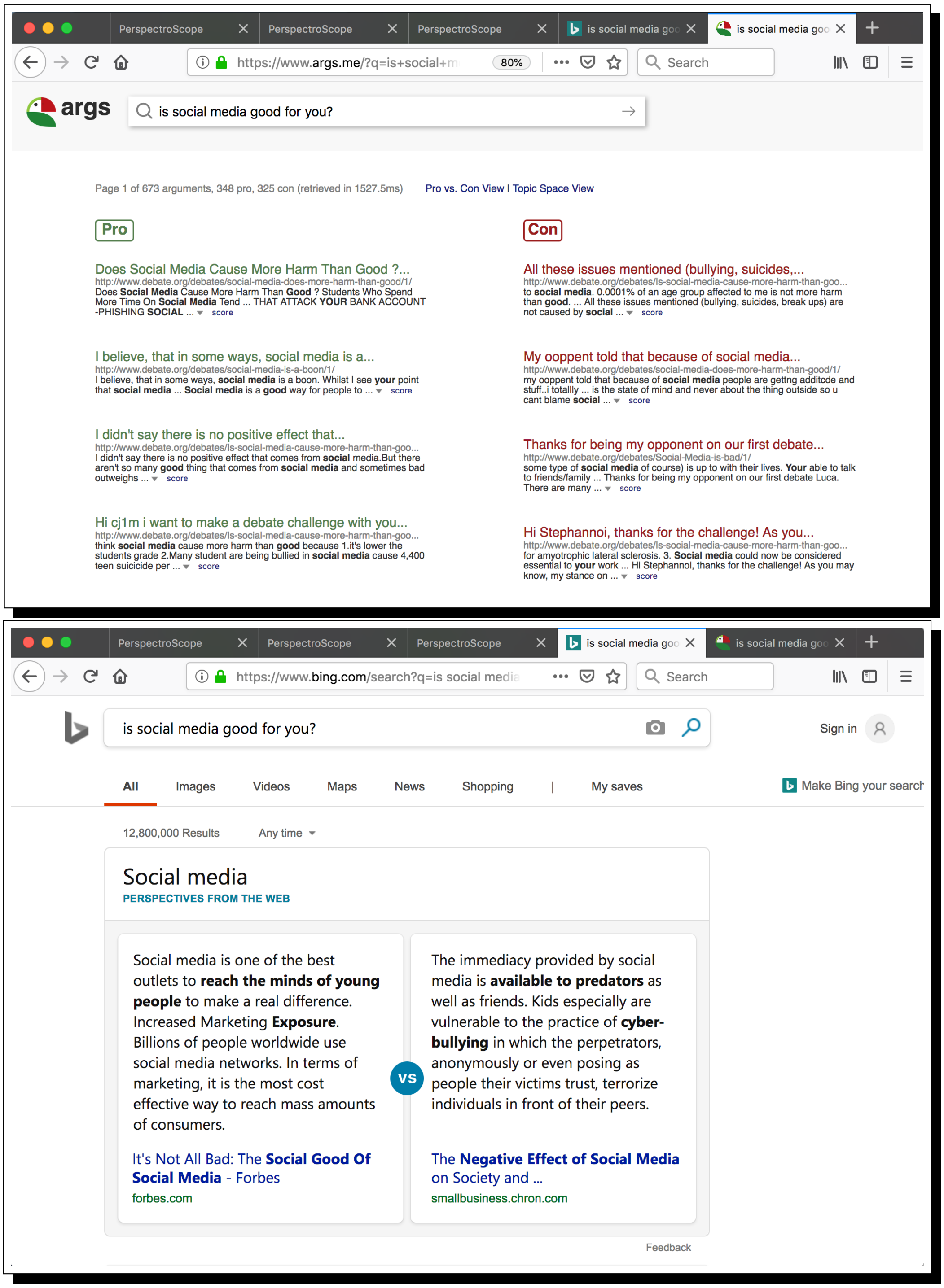

Figure 4: Related work: `args.me` an argument retrieval engine using arguments extracted from debate websites; `bing.com` search engine showing contrasting views on a debate topic.

4 Conclusion and Future Work

We have presented PERSPECTROSCOPE, a powerful interface for exploring different perspectives to discussion-worthy claims. The system is built with a combination of retrieval engines and learned classifiers to create a good balance between speed and quality. Our system is designed with the mindset of being able to get feedback from users of the system.

While this work offers a good step towards a higher quality and flexible interface, there are many issues and limitations that are not addressed here and are opportunities for future work. For instance, the system provided here does not provide any guarantees in terms of the *exhaustiveness* of the perspectives in the world, or levels of expertise and trustworthiness of the identified evidence. Moreover, any classifier trained on some annotated data (such as what we used here) could potentially contain hidden biases that might not be easy to see. We hope that some of these challenges and limitations will be addressed in future work.

Acknowledgments

This work was supported in part by a gift from Google and by Contract HR0011-15-2-0025 with the US Defense Advanced Research Projects Agency (DARPA). The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

S. Chen, D. Khashabi, W. Yin, C. Callison-Burch, and D. Roth. 2019. Seeing things from a different angle: Discovering diverse perspectives about claims. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 542–557.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, pages 4171–4186.

M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 1996 Conference on Knowledge Discovery & Data Mining*, pages 226–231.

K. S. Hasan and V. Ng. 2014. Why are you taking this stance? identifying and classifying reasons in ideological debates. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 751–762.

X. Hua and L. 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*, pages 203–208.

R. Levy, B. Bogin, S. Gretz, R. Aharonov, and N. Slonim. 2018. Towards an argumentative content search engine using weak supervision. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2066–2081.

M. Nadeem, W. Fang, B. Xu, M. Mohtarami, and J. Glass. 2019. Fakta: An automatic end-to-end fact checking system. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 78–83.

C. Stab, J. Daxenberger, C. Stahlhut, T. Miller, B. Schiller, C. Tauchmann, S. Eger, and I. Gurevych. 2018a. Argumentext: Searching for arguments in heterogeneous sources. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 21–25.

C. Stab, T. Miller, B. Schiller, P. Rai, and I. Gurevych. 2018b. Cross-topic argument mining from heterogeneous sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3664–3674.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a Large-scale Dataset for Fact Extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 809–819.

H. Wachsmuth, M. Potthast, K. Al Khatib, Y. Ajjour, J. Puschmann, J. Qu, J. Dorsch, V. Morari, J. Bevendorff, and B. Stein. 2017. Building an argument search engine for the web. In *Workshop on Argument Mining*.

H. Wachsmuth, S. Syed, and B. Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Volume 1*, pages 241–251.