Extractive Snippet Generation for Arguments

Milad Alshomary Nick Düsterhus Henning Wachsmuth

Computational Social Science Group, Department of Computer Science, Paderborn University, Paderborn, Germany
milad.alshomary@upb.de nduester@mail.upb.de henningw@upb.de

ABSTRACT

Snippets are used in web search to help users assess the relevance of retrieved results to their query. Recently, specialized search engines have arisen that retrieve pro and con arguments on controversial issues. We argue that standard snippet generation is insufficient to represent the core reasoning of an argument. In this paper, we introduce the task of generating a snippet that represents the main claim and reason of an argument. We propose a query-independent extractive summarization approach to this task that uses a variant of PageRank to assess the importance of sentences based on their context and argumentativeness. In both automatic and manual evaluation, our approach outperforms strong baselines.

1 INTRODUCTION

General-purpose search engines present results along with short text *snippets* that help users to assess the relevance of the underlying web page to their queries [5]. These snippets boost the usability of search engines [14], and have been studied extensively. Usually, a snippet shows a representative excerpt of the web page's content, ideally including all query terms [5]. This is a suitable compromise for the varying search goals that users may have [17].

Recently, there is a growth of research on the computational analysis of natural language arguments [19], where an argument is seen as a combination of claims and supporting reasons, along with descriptive information. As part of this research, more attention is given to argument retrieval from the web [6, 16], and specialized search engines have been proposed which aim to give an efficient overview of the best arguments on a controversial issue queried [18, 20]. For this specific search scenario, we hypothesize that general-purpose snippets are not sufficient. For illustration, Figure 1 shows an exemplary result from our argument search engine *args.me* [20]. A general-purpose snippet would likely be based on the argument's beginning, which contains the query term multiple times. However, the main claim and reason of the argument follow later, shown underlined in the figure. Hence, we argue that specific snippets are needed for arguments. To our knowledge, such argument snippet generation is not yet touched upon in existing research.

In this paper, we lay the groundwork to tackle argument snippet generation. As detailed in Section 3, our working definition is that a good argument snippet represents the main claim and the main reason to support it, thus capturing the core of the argument. We provide a first evaluation dataset based on 10 arguments each for the top-10 queries sent to args.me [1]. For each argument, two experts agreed on the two sentences that form the best snippet according to our definition.[1] The resulting task hence is to automatically generate the best two-sentence snippet for a given argument.

Query abortion

Argument The Supreme Court decided that states can't outlaw *abortion* because Prohibiting *abortion* is a violation of the 14th Amendment, according to the Court, and the constitution. Outlawing *abortion* is taking away a human right given to women. In reality, a fetus is just a bunch of cells. It has not fully developed any vital organs like lungs. This means that an *abortion* is not murder, it is just killing of cells in the wound. If the child has no organs developed that would be vital for the baby to survive outside the wound, than having an *abortion* is not murder.

Figure 1: Example argument for the query "abortion". The underlined sentences (ordered as given) form the best two-sentence snippet according to our annotators.

For reasons discussed below, we propose an *unsupervised extractive summarization approach* to generate snippets (Section 4). Our core idea is that the importance of a claim or reason within an argument depends on how often it is referred to in the argument's context (e.g., the covering web page). To operationalize this idea, we propose a graph-based method that ranks an argument's sentences by their centrality in the graph. Nodes represent the sentences in the context, and edges are created between similar sentences (see Figure 2). In line with related work [7], we then employ a variant of PageRank [21] to obtain a score for each sentence. A unique feature of our approach is, though, that we bias PageRank towards argumentative language, employing a discourse lexicon [11].

In empirical experiments on the new dataset (Section 5), our approach outperforms strong extractive summarization baselines, *LexRank* [7] and *BertSum* [13], in picking the best snippet sentences. Moreover, in two studies we asked three users to rank candidate snippets by their representativeness [12] and readability [10]. First, we evaluated all extractive approaches against the expert snippets. Our approach excelled in representativeness and ranked second for readability. Then, we compared the snippets of our approach to those returned by args.me and to query-dependent snippets created by *Lucene*. In the clear majority of cases, we performed best in both in representativeness (60%) and in readability (58%), supporting our hypothesis that general-purpose snippets are insufficient in argument search. The main contributions of this paper are:

(1) The definition of the new task of argument snippet generation, along with a first dataset for evaluation.

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07.
https://doi.org/10.1145/3397271.3401186

[1] The annotated dataset can be found under: https://github.com/webis-de/SIGIR-20.

(2) An extractive argument snippet generation approach that seeks for the argumentatively most representative sentences.
(3) Empirical evidence that general-purpose snippet generation approaches are insufficient for arguments.

2 RELATED WORK

Search engines users have different goals that can generally be classified into *navigational*, *informational*, and *resource* [17]. Multiple works stress the impact of snippets on search engine usability to achieve these goals [14]. In argument search, the goal is mostly informational but undirected, raising the need for an efficient overview [18, 20]. This gives snippets special importance.

In general, snippet generation can be understood as a summarization task [5]. Accordingly, snippets may be generated either in an extractive or in an abstractive way [5]. While a recent user study suggests that both are equally appreciated [4], we restrict our view to extractive summaries here, as they tend to be more intuitive and reliable [7]. Gholamrezazadeh et al. [8] comprehensively surveyed summarization, distinguishing statistical, clustering, rhetorical, and graph methods. Inspired by Erkan and Radev [7], we adopt a graph-based approach here. In particular, we rely on PageRank, but we bias it towards the argumentative nature of sentences.

In the context of search, snippets may be generated either query-dependent (with regard to a user's query) or query-independent [5]. By concept, only the former allows optimizing a snippet towards different issues that an argument might be relevant to (e.g., *abortion* as opposed to *state sovereignty* for the example in Figure 1). However, Ajjour et al. [1] observed that most queries sent to argument search engines are one or two words short, directly addressing the issue of interest. As long as the retrieved arguments convey a stance towards the issue, a query-independent snippet should hence be suitable. We thus generate such snippets with our approach.

Computational research on argumentation so far mainly focuses on the mining of claims, reasons, and their relations from text [19]. Search-related approaches retrieve argumentative sentences, and rank them by argumentative support [2]. Also, argument relevance has been assessed [16], partly using PageRank [21]. Unlike these works, we *summarize* arguments in snippets. Summarization has been used to find the main points in debates [22], but to our knowledge not to boil down an argument to its core. This is the gap we aim to fill with argument snippet generation.

3 TASK AND DATA

The assumption behind argument search is that users who aim to form a stance on a controversial issue need an efficient overview of the most relevant arguments [18, 20]. Ideally, each argument consists of a (main) claim that conveys a stance on the issue or its aspects, along with a (main) reason supporting the claim [19]. Practically, argumentative texts may phrase multiple claims and reasons, spread them over multiple sentences, add non-argumentative background information, all combined in various ways. To understand the core of an argument, a user needs to identify the actual inference from reason to claim, irrespective of the phrasing. We think that an argument snippet should support this process.

Accordingly, our working definition is that a good argument snippet represents the main claim and the main reason supporting it in a short summary.[2] Since we expect that claim and reason can be expressed in a single sentence each and since two sentences roughly match what fits into the typical length of a search engine snippet, we restrict our view to two-sentence snippets, and we define the argument snippet generation task accordingly as:

> *Given a natural language argument, generate a two-sentence snippet that best represents the argument's main claim and main reason.*

For evaluation, we provide a first benchmark dataset with ground-truth snippets for a sample of arguments: To this end, we retrieved arguments for the 10 queries most often submitted to *args.me* [1]. All arguments come from debate-like web pages (see Figure 1 for an example). Each has a stance towards an issue-like conclusion and a debate identifier. For each query, we took the top five pro and top five con arguments and filtered out all trivial cases, i.e., those with maximum two sentences. The length of the remaining 73 arguments ranges from 3 to 84 sentences with a median of 9. We asked two human experts to select the two sentences from each argument that, in their given ordering, define the most representative snippet according to our working definition. In 77% of the cases, the experts agreed on at least one sentence, with a Cohen's κ agreement of 0.50. Disagreement cases were resolved in on-site discussion between them, which worked out well in all cases. We randomly split the final dataset into 23 arguments for development and 50 for testing.

4 APPROACH

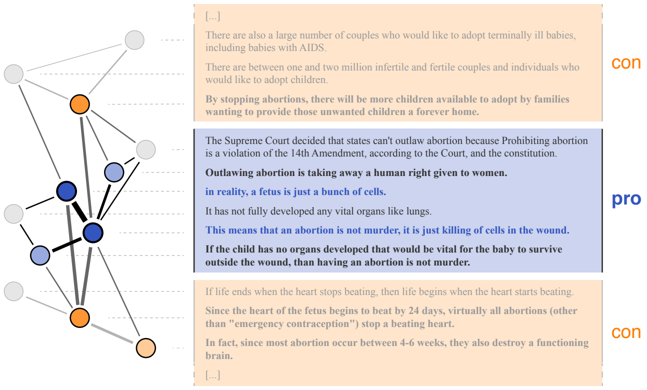
Our approach to the given task first ranks sentences by their importance of representing the core of an argument then it returns the two top-ranked sentences. In particular, we hypothesize that the importance is influenced by two criteria: *centrality in context* and *argumentativeness.* On one hand, the main claim and reason of an argument should likely be often referred to across arguments from the same context. Such context could be arguments in a debate on the same web page or a cluster of arguments addressing the same issue. On the other hand, sentences that contain claims or reasons are likely to contain argumentative language. To operationalize these criteria, we propose a graph-based approach following Erkan and Radev [7] but utilizing sentence embeddings to capture both syntactic and semantic features of sentences. Moreover, we explicitly bias the model towards argumentative sentences.

Concretely, given an argument, we first retrieve other arguments from the same context (in our experiments, we use all arguments from the same web page). Second, for each sentence s in the given and in the retrieved arguments, an embedding is generated and an argumentative score $arg(s)$ is computed. The resulting graph covers all sentences (represented by their embedding) as nodes, and the similarity $sim(s_i, s_j)$ of the embeddings represents the edge weight between s_i and s_j. Finally, we apply the PageRank [15] algorithm to generate an importance score $P(s_i)$ for each sentence s_i:

$$P(s_i) \;=\; (1-d) \cdot \sum_{s_j \neq s_i} \frac{sim(s_i, s_j)}{\sum_{s_k \neq s_j} sim(s_j, s_k)} P(s_j) \;+\; d \cdot \frac{arg(s_i)}{\sum_{s_k} arg(s_k)}$$

As captured by the equation, $P(s_i)$ is a sum of two parts, weighted by a damping factor d. The first part reflects the centrality of s_i in

[2] The hypothesis underlying this definition is up to further analysis that is beyond the scope here. To some extent, however, our experiments below provide support for it.



[...]

There are also a large number of couples who would like to adopt terminally ill babies, including babies with AIDS.

There are between one and two million infertile and fertile couples and individuals who would like to adopt children.

By stopping abortions, there will be more children available to adopt by families wanting to provide those unwanted children a forever home.

The Supreme Court decided that states can't outlaw abortion because Prohibiting abortion is a violation of the 14th Amendment, according to the Court, and the constitution.

Outlawing abortion is taking away a human right given to women.

in reality, a fetus is just a bunch of cells.

It has not fully developed any vital organs like lungs.

This means that an abortion is not murder, it is just killing of cells in the wound.

If the child has no organs developed that would be vital for the baby to survive outside the wound, than having an abortion is not murder.

If life ends when the heart stops beating, then life begins when the heart starts beating.

Since the heart of the fetus begins to beat by 24 days, virtually all abortions (other than "emergency contraception") stop a beating heart.

In fact, since most abortion occur between 4-6 weeks, they also destroy a functioning brain.

[...]

Figure 2: Illustration of our sentence scoring graph method for a pro argument on a web page discussing abortion. Nodes with black borders represent argumentative sentences (bold text). Edge thickness reflects similarity and bias towards argumentativeness (very thin edges are omitted for a lean visualization). The more saturated a node, the higher the score.

its context, computed with respect to the importance $P(s_j)$ of other sentences s_j and their similarity to s_i. The second part represents a normalized bias towards the argumentativeness nature of s_i, which can be computed by utilizing argument mining techniques. Figure 2 illustrates the influence of similarity and argumentativeness. As in the original PageRank, we start with equal scores for all sentences and iteratively update them until near-convergence. We then rank all sentences of the given argument by score and generate a snippet from the two top-ranked sentences. We concatenate them in their original ordering, in order to preserve their intrinsic coherence.

5 EVALUATION

We now present empirical experiments, conducted to evaluate our approach in generating representative two-sentence snippets on the dataset from Section 3, in light of our hypothesis that general-purpose snippets are insufficient for arguments.

Our Approach. We model the context of an argument by all other arguments in the same debate (Section 3).[3] Sentences are embedded using the universal sentence embedding model [3], and cosine similarity is computed between them. To compute the *argumentative score*, we use a discourse lexicon, constructed from discourse markers and claim words [11]. We evaluate two variants: *(a) our approach (centrality)* considers only the centrality of a sentence in its debate context ($d = 0$), and *(a^*) our approach (centrality + argumentativeness)* computes the argumentative score of a sentence as the frequency of discourse markers in it ($d = 0.15$, default value).[4]

Baselines. We compare our snippet generation approach to *(b_1) random sentence selection*, in order to assess what gains we achieve. In addition, we consider two extractive summarization baselines: On one hand, we use the Python implementation of the unsupervised graph-based method *(b_2) LexRank* [7]. As our approach, we apply it to all sentences of all arguments from the same debate. We create an unweighted edge when the similarity of two sentences exceeds 0.1. For further optimization, *(b_2^*) LexRank + weighted similarity* uses the similarities themselves as the weights of all respective edges. On the other hand, we employ the supervised method *(b_3) BertSum* [13] to extract those two sentences as the snippet that most resemble the argument's conclusion, since the latter may retain representative information. We trained *BertSum* on the dataset from [22], which contains pairs of premises and conclusion from online debates, considering the conclusion as the summary of its premises. Since these conclusions are not always excerpts from the premises, we modified b_3 to *(b_3^*) BertSum + optimized towards snippet extraction*, such that the summary is given by the two sentences that most overlap with the conclusion in content tokens, and trained it accordingly.

Table 1: Automatic evaluation: Accuracy of all evaluated snippet generation baselines and our approach variations in matching the ground-truth snippets of the given test set.

#	Approach	Accuracy
b1	Random sentence selection	27%
b2	LexRank	27%
b2*	LexRank + weighted similarity	36%
b3	BertSum	33%
b3*	BertSum + optimized towards snippet extraction	40%
a	Our Approach (centrality)	43%
a*	Our Approach (centrality + argumentativeness)	**44%**

Automatic Evaluation. We computed the accuracy of each approach in selecting the ground-truth snippets' sentences, averaged over all arguments in the dataset: For each argument, the accuracy is either 0 (no selected sentence correct), 0.5 (one correct), or 1 (both correct). Table 1 shows the results. While *LexRank* does not improve over *random sentence selection* (both 27%), the modification b_2^* improves it to 36%, and the optimized *BertSum* (b_3^*) even achieves 40%. *Our approach (centrality)* already achieves higher accuracy than all baselines (43%). Encoding argumentativeness of a sentence (a^*) further increases accuracy, only slightly though (44%). We expect that the gain would be larger on datasets with fewer argumentative sentences, and with refined argument mining techniques, but we leave both to future work. Here, we conclude that modeling the centrality and argumentativeness of an argument's sentences turns out most successful in mimicking the ground-truth snippets.

Manual Evaluation. To assess the quality of the generated snippets, we conducted two annotation studies, each with an independent set of three university students that have background on search engines and argumentation. Following literature, we mainly consider a snippet's *representativeness* [12] in terms of capturing the core information of the corresponding argument. In addition, we include *readability* [10] as another quality dimension, here meaning how coherent the two sentences of a snippet are on their own.

The first study compared our approach to the two best baselines and to the expert snippets. After explaining the task and quality dimensions, we showed the four competing snippets in random order for all 50 test arguments, and asked the annotators to rank the

[3] Other methods may consider the context as a cluster of arguments on the same topic.

[4] We also experimented with argument mining, training an SVM over web argument data [9] to classify sentences as *claim*, *premise*, or *non-argumentative*. However, the output was of limited reliability and did not improve over using the discourse lexicon.

Table 2: Manual evaluation: Mean readability and representativeness rank (lower is better) and proportion of best ranks (higher is better, multiple best ranks possible) of the test set snippets of selected approaches and of the expert snippets.

	Approach	**Readability**		**Representativeness**	
#		**Mean Rank**	**% Rank 1**	**Mean Rank**	**% Rank 1**
b2*	LexRank + w.s.	2.57	28%	2.47	28%
b3*	BertSum + o.t.s.e.	**2.15**	**46%**	2.43	26%
a*	Our Approach (c.+a.)	2.50	26%	**1.95**	**44%**
	Expert snippets	1.71	66%	1.66	60%

snippets according to both dimensions. To avoid bias, no training was done before. The mean pairwise inter-annotator agreement in terms of the rank-correlation measure Spearman's ρ was 0.52 for representativeness and 0.36 for readability, indicating general agreement but notable subjectivity in the given task. To give each annotator equal importance, we thus computed the mean ranks over all annotators. As Table 2 presents, *our approach* clearly outperformed the others in terms of representativeness, being best in 44% of the cases and achieving a mean rank of 1.95. However, the readability of its snippets was judged worse than for *BertSum.* A reason may be that our approach tends to favor long sentences by concept. Besides, snippets generated by the experts proved best, underlining that our working definition from Section 3 is reasonable.

In the second study, we assessed whether our approach improves over approaches from practice. For this, we compared to the built-in snippet generation of *Lucene*, which is *query-dependent*: it selects text spans overlapping with query tokens (we reused the queries from Section 3). In addition, we evaluated the current snippets of *args.me*, which just show the beginning of arguments. All snippets were truncated after 225 characters, to mimic a real user-interface situation. We used the same setting as in the first user study but with different students to avoid bias. Spearman's ρ was 0.33 for representativeness and 0.36 for readability. Table 3 shows that *our approach* again performs best in terms of representativeness, and is on par with the readability of *args.me.* The fact that our approach produces the most representative snippets in 60% of the cases provides empirical evidence for the need to address argument snippet generation as a special task, and highlights the limited of (at least standard) query-dependent snippet generation in argument search scenarios. We hence plan to integrate our approach into *args.me.*

6 CONCLUSION

Motivated by the need for specialized snippets in argument search, this work has introduced argument snippet generation as a new task, along with a first benchmark dataset. We have introduced a graph-based extractive summarization approach that captures the importance of a sentence based on its *centrality in context* and its *argumentativeness.* In data-driven experiments as well as two user studies, we have provided evidence that our approach outperforms strong extractive baselines as well as standard query-dependent snippet generation. Our findings lay the groundwork for research on better approaches to generating argument snippets in the context of both general and specialized argument search engines.

Table 3: Second manual evaluation: Same as Table 2, but for snippets generated by the query-dependent Lucene algorithm, the search engine args.me, and our approach.

	Readability		**Representativeness**	
	Mean Rank	**% Rank 1**	**Mean Rank**	**% Rank 1**
Lucene (query-dependent)	2.06	28%	2.17	22%
Args.me	**1.52**	48%	1.77	50%
Our Approach (c.+a.)	1.60	**58%**	**1.69**	**60%**

REFERENCES

[1] Yamen Ajjour, Henning Wachsmuth, Johannes Kiesel, Martin Potthast, Matthias Hagen, and Benno Stein. 2019. Data Acquisition for Argument Search: The args.me Corpus. In *Proceedings of the KI.* 48–59.

[2] 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 ECIR.* 129–141.

[3] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal Sentence Encoder. *arXiv preprint arXiv:1803.11175* (2018).

[4] Wei-Fan Chen, Matthias Hagen, Benno Stein, and Martin Potthast. 2018. A User Study on Snippet Generation: Text Reuse vs. Paraphrases. In *Proceedings of the 41st SIGIR.* ACM, 1033–1036.

[5] Bruce Croft, Donald Metzler, and Trevor Strohman. 2009. *Search Engines: Information Retrieval in Practice* (1st ed.). Addison-Wesley.

[6] Lorik Dumani and Ralf Schenkel. 2019. A Systematic Comparison of Methods for Finding Good Premises for Claims. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 957–960.

[7] Günes Erkan and Dragomir R Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research* 22 (2004), 457–479.

[8] Saeedeh Gholamrezazadeh, Mohsen Amini Salehi, and Bahareh Gholamzadeh. 2009. A Comprehensive Survey on Text Summarization Systems. In *Proceedings of the 2nd CSA.* 1–6.

[9] Ivan Habernal and Iryna Gurevych. 2017. Argumentation Mining in User-Generated Web Discourse. *Computational Linguistics* 43, 1 (2017), 125–179.

[10] Tapas Kanungo and David Orr. 2009. Predicting the readability of short web summaries. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining.* 202–211.

[11] Ran Levy, Shai Gretz, Benjamin Sznajder, Shay Hummel, Ranit Aharonov, and Noam Slonim. 2017. Unsupervised corpus–wide claim detection. In *Proceedings of the 4th Workshop on Argument Mining.* 79–84.

[12] Shao Fen Liang, Siobhan Devlin, and John Tait. 2006. Evaluating web search result summaries. In *Proceedings of the 28th ECIR.* Springer, 96–106.

[13] Yang Liu. 2019. Fine-tune BERT for Extractive Summarization. *arXiv preprint arXiv:1903.10318* (2019).

[14] Mari-Carmen Marcos, Ferran Gavin, and Ioannis Arapakis. 2015. Effect of Snippets on User Experience in Web Search. In *Proceedings of the 16th HCI.* 47.

[15] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web.* Technical Report. Stanford InfoLab.

[16] Martin Potthast, Lukas Gienapp, Florian Euchner, Nick Heilenkötter, Nico Weidmann, Henning Wachsmuth, Benno Stein, and Matthias Hagen. 2019. Argument Search: Assessing Argument Relevance. In *Proceedings of the 42nd SIGIR.* 1117–1120.

[17] Daniel E Rose and Danny Levinson. 2004. Understanding User Goals in Web Search. In *Proceedings of the 13th WWW.* 13–19.

[18] Christian Stab, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. 2018. ArgumenText: Searching for Arguments in Heterogeneous Sources. In *Proceedings of the 2018 NAACL: Demonstrations.* 21–25.

[19] Manfred Stede and Jodi Schneider. 2018. *Argumentation Mining.* Number 40 in Synthesis Lectures on Human Language Technologies. Morgan & Claypool.

[20] 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.* 49–59.

[21] Henning Wachsmuth, Benno Stein, and Yamen Ajjour. 2017. "PageRank" for Argument Relevance. In *Proceedings of the 15th EACL.* 1117–1127.

[22] Lu Wang and Wang Ling. 2016. Neural Network-Based Abstract Generation for Opinions and Arguments. In *Proceedings of the 2016 NAACL.* 47–57.