Retrieval of the Best Counterargument without Prior Topic Knowledge

Henning Wachsmuth
Paderborn University
Computational Social Science Group
henningw@upb.de

Shahbaz Syed and **Benno Stein**
Bauhaus-Universität Weimar
Faculty of Media, Webis Group
<first>.<last>@uni-weimar.de

Abstract

Given any argument on any controversial topic, how to counter it? This question implies the challenging retrieval task of finding the best counterargument. Since prior knowledge of a topic cannot be expected in general, we hypothesize the best counterargument to invoke the same aspects as the argument while having the opposite stance. To operationalize our hypothesis, we simultaneously model the similarity and dissimilarity of pairs of arguments, based on the words and embeddings of the arguments' premises and conclusions. A salient property of our model is its independence from the topic at hand, i.e., it applies to arbitrary arguments. We evaluate different model variations on millions of argument pairs derived from the web portal *idebate.org*. Systematic ranking experiments suggest that our hypothesis is true for many arguments: For 7.6 candidates with opposing stance on average, we rank the best counterargument highest with 60% accuracy. Even among all 2801 test set pairs as candidates, we still find the best one about every third time.

1 Introduction

Many controversial topics in real life divide us into opposing camps, such as whether to ban guns, who should become president, or what phone to buy. When being confronted with arguments against our stance, but also when forming own arguments, we need to think about how they could best be countered. Argumentation theory tells us that — aside from ad-hominem attacks — a counterargument denies either an argument's premises, its conclusion, or the reasoning between them (Walton, 2009). Take the following argument in favor of the right to bear arms from the web portal idebate.org:

Argument *"Gun ownership is an integral aspect of the right to self defence.* (conclusion) *Law-abiding citizens deserve the right to protect their families in their own homes, especially if the police are judged incapable of dealing with the threat of attack. [...]"* (premise)

While the conclusion seems well-reasoned, the web portal directly provides a counter to the argument:

Counterargument *"Burglary should not be punished by vigilante killings of the offender. No amount of property is worth a human life. Perversely, the danger of attack by homeowners may make it more likely that criminals will carry their own weapons. If a right to self-defence is granted in this way, many accidental deaths are bound to result. [...]"*

As in this example, we observe that a counterargument often takes on the aspects of the topic invoked by the argument, while adding a new perspective to its conclusion and/or premises, conveying the opposite stance. Research has tackled the stance of argument units (Bar-Haim et al., 2017) as well as the attack relations between arguments (Cabrio and Villata, 2012). However, existing approaches learn the interplay of aspects and topics on training data or infer it from external knowledge bases (details in Section 2). This does not work for topics unseen before. Moreover, to our knowledge, no work so far aims at actual *counter*arguments.

This paper studies the task of automatically finding the best counterargument to any argument. In the general case, we cannot expect prior knowledge of an argument's topic. Following the observation above, we thus just hypothesize the best counterargument to invoke the same aspects as the argument while having the opposite stance. Figure 1 sketches how we operationalize the hypothesis. In particular, we simultaneously model the topic similarity and stance *dis*similarity of a candidate counterargument

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pages 241–251
Melbourne, Australia, July 15 - 20, 2018. ©2018 Association for Computational Linguistics

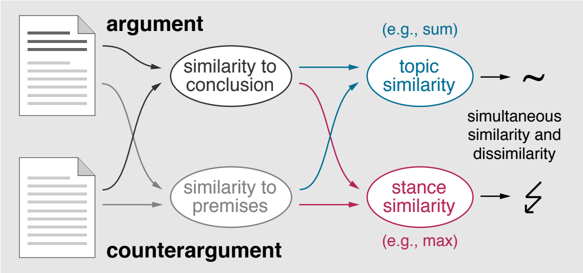

Figure 1: Modeling the simultaneous similarity and dissimilarity of a counterargument to an argument.

to the argument. Both are inferred — in different ways — from the similarities to the argument's conclusion and premises, since it is unclear in advance, whether either of these units or the reasoning between them is countered. Thereby, we find the most dissimilar among the most similar arguments.

To study counteraguments, we provide a new corpus with 6753 argument-counterargument pairs, taken from 1069 debates on idebate.org, as well as millions of false pairs derived from them. Given the corpus, we define eight retrieval tasks that differ in the types of candidate counterarguments. Based on the words and embeddings of the arguments, we develop similarity functions that realize the outlined model as a ranking approach. In systematic experiments, we evaluate the different building blocks of our model on all defined tasks.

The results suggest that our hypothesis is true for many arguments. The best model configuration improves common word and embedding similarity measures by eight to ten points accuracy in all tasks. Inter alia, we rank 60.3% of the best counterarguments highest when given all arguments with opposite stance (7.6 on average). Even with all 2801 test arguments as candidates, we still achieve 32.4% (and a mean rank of 15), fitting the intuition that off-topic arguments are easier to discard. Our analysis reveals notable gaps across topical themes though.

Contributions We believe that our findings will be important for applications such as automatic debating technologies (Rinott et al., 2015) and argument search (Wachsmuth et al., 2017b). To summarize, our main contributions are:

- A large corpus for studying multiple counterargument retrieval tasks (Sections 3 and 4).
- A topic-independent approach to find the best counterargument to any argument (Section 5).
- Evidence that many counterarguments can be found without topic knowledge (Section 6).

The corpus as well as the Java source code for reproducing the experiments are available at `http://www.arguana.com`.

2 Related Work

Counterarguments rebut arguments. In the theoretical model of Toulmin (1958), a rebuttal in fact does not attack the argument, but it merely shows exceptions to the argument's reasoning. Govier (2010) suggests to rather speak of counterconsiderations in such cases. Unlike Damer (2009), who investigates *how to attack* several kinds of fallacies, we are interested in *how to identify attacks*. We focus on those that target arguments, excluding personal (ad-hominem) attacks (Habernal et al., 2018).

Following Walton (2006), an argument can be attacked in two ways: one is to question its validity — not meaning that its conclusion must be wrong. The other is to rebut it with a *counterargument* that entails the opposite conclusion, often by revisiting aspects or introducing new ones. This is the type of attack we study. As Walton (2009) details, rebuttals may target an argument's premises or conclusion, or they may undercut the reasoning between them.

Recently, the computational analysis of natural language argumentation is receiving much attention. Most research focuses on argument mining, ranging from segmenting a text into argument units (Ajjour et al., 2017), over identifying unit types (Rinott et al., 2015) and roles (Niculae et al., 2017), to classifying argument schemes (Feng and Hirst, 2011) and relations (Lawrence and Reed, 2017). Some works detect counterconsiderations in a text (Peldszus and Stede, 2015) or their absence (Stab and Gurevych, 2016). Such considerations make arguments more balanced (see above). In contrast, we seek for arguments that defeat others.

Many approaches mine attack relations between arguments. Some use deep learning to find attacks in discussions (Cocarascu and Toni, 2017). Closer to this paper, others determine them in a given set of arguments, using textual entailment (Cabrio and Villata, 2012) or a combination of markov logic and stance classification (Hou and Jochim, 2017). In principle, any attacking argument denotes a counterargument. Unlike previous work, however, we aim for the *best* counterargument to an argument.

Classifying the stance of a text towards a topic (pro or con) generally defines an alternative way of addressing counterarguments. Sobhani et al. (2015) specifically classify health-related arguments using supervised learning, while we do not expect to have prior topic knowledge. Bar-Haim et al. (2017) approach the stance of claims towards open-domain topics. Their approach combines aspect-based sentiment with external relations between aspects and topics from Wikipedia. As such, it is in fact limited to the topics covered there. Our model applies to arbitrary arguments and counterarguments.

We need to identify only whether arguments oppose each other, not their actual stance. Similarly, Menini et al. (2017) classify only the *disagreement* of political texts. Part of their approach is to detect topical key aspects in an unsupervised manner, which seems useful for our purposes. Analogously, Beigman Klebanov et al. (2010) study differences in vocabulary choice for the related task of perspective classification, and Tan et al. (2016) find that the best way to persuade opinion holders in the *Change my view* forum on reddit.com is to use dissimilar words. As we report later, however, our experiments did not show such results for the argument-counterargument pairs we deal with.

The goal of persuasion reveals the association of counterarguments to argumentation quality. Many quality criteria have been assessed for arguments, surveyed in (Wachsmuth et al., 2017a). In the study of Habernal and Gurevych (2016), one reason annotators gave for why an argument was more convincing than another was that it tackled flaws in the opposing view. Zhang et al. (2016) even found that debate winners tend to counter opposing arguments rather than focusing on their own arguments.

Argument quality assessment is particularly important in retrieval scenarios. Existing approaches aim to retrieve documents that contain many claims (Roitman et al., 2016) or that provide most support for their claims (Braunstain et al., 2016). In Wachsmuth et al. (2017c), we adapt PageRank to argumentative relations, in order to assess argument relevance objectively. While our search engine *args* for arguments on the web still uses content-based relevance measures in its first version (Wachsmuth et al., 2017b), its long-term idea is to rank the best arguments highest.[1] The model present in this work finds the best *counter*arguments, but it is meant to be integrated into *args* at some point.

Like here, *args* uses idebate.org arguments. Others take data from that portal for studying support (Boltužić and Šnajder, 2014) or for the distant supervision of argument mining (Al-Khatib et al., 2016). Our corpus is not only larger, though, but it is the first to utilize a unique feature of idebate.org: the explicit specification of counterarguments.

3 The ArguAna Counterargs Corpus

This section introduces our *ArguAna Counterargs corpus* with argument-counterargument pairs, created automatically from the structure of idebate.org. The corpus is freely available at `http://www.arguana.com/data`. We also provide the code to replicate the construction process.

3.1 The Web Portal idebate.org

On the *portal* idebate.org, diverse controversial topics of usually rather general interest are discussed in *debates*, subsumed under 15 *themes*, such as “economy” and “health”. Each debate has a title capturing a thesis on a topic, such as “This House would limit the right to bear arms”, followed by an introductory text, a set of mostly elaborated and well-written *points* that have a pro or a con stance towards the thesis, and a bibliography.

A specific feature of idebate.org is that virtually every point comes along with a *counter* that immediately attacks the point and its stance. Both points and counters can be seen as *arguments*. While a point consists of a one-sentence claim (the argument’s *conclusion*) and a few sentences justifying the claim (the *premise(s)*), the counter’s (opposite) conclusion remains implicit.

All arguments on the portal are established by a community with the goal of showing both sides of a topic in a balanced manner. We therefore assume each counter to be the best counterargument available for the respective point, and we use all resulting *true argument pairs* as the basis of our corpus. Figure 2 illustrates the italicized concepts, showing the structure of idebate.org. An example argument pair has been discussed in Section 1.

3.2 Corpus Construction

We crawled all debates from idebate.org that follow the portal’s theme-guided folder structure (last access: January 30, 2018). From each debate, we extracted the thesis, the introductory text, all points and counters, the bibliography, and some metadata. Each was stored separately in one plain text file, and we also created a file with the entire debate in its original order. Only points and counters are used in our experiments in Section 6. The underlying experiment settings are described in Section 4.

[1] Argument search engine *args*: `http://args.me`

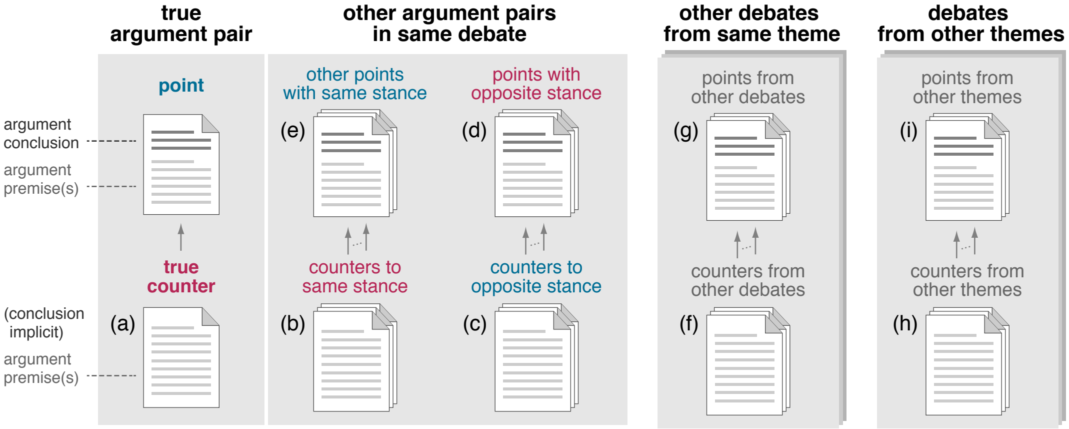
true argument pair
other argument pairs in same debate
other debates from same theme
debates from other themes
point
other points with same stance
points with opposite stance
points from other debates
points from other themes
argument conclusion
argument premise(s)
(e)
(d)
(g)
(i)
true counter
counters to same stance
counters to opposite stance
counters from other debates
counters from other themes
(conclusion implicit)
argument premise(s)
(a)
(b)
(c)
(f)
(h)

Figure 2: Structure of idebate.org for one true argument pair in our corpus. Colors denote matching stance; we assume arguments from other debates to have no stance towards a point. Points have a conclusion and premises, counters only premises. (a)–(i) are used in Section 4 to specify the candidates in different tasks.

Theme	Debates	Points	Counters
Culture	46	278	278
Digital freedoms	48	341	341
Economy	95	590	588
Education	58	382	381
Environment	36	215	215
Free speech debate	43	274	273
Health	57	334	333
International	196	1315	1307
Law	116	732	730
Philosophy	50	320	320
Politics	155	982	978
Religion	30	179	179
Science	41	271	269
Society	75	436	431
Sport	23	130	130
Training set	644	4083	4065
Validation set	211	1290	1287
Test set	214	1406	1401
counterargs-18	**1069**	**6779**	**6753**

Table 1: Distribution of debates, points, and counters over the themes in the *counterargs-18* corpus. The bottom rows show the size of the datasets.

3.3 Corpus Statistics

Table 1 lists the number of debates crawled for each theme, along with the numbers of points and counters in the debates. The 26 found points without a counter are included in the corpus, but we do not use them in our experiments.

In total, the ArguAna Counterargs corpus consists of 1069 debates with 6753 points that have a counter. The mean length of points is 196.3 words, whereas counters span only 129.6 words, largely due to the missing explicit conclusion. To avoid exploiting this corpus bias, no approach in our experiments captures length differences.

3.4 Datasets

We split the corpus into a training set, consisting of the first 60% of all debates of each theme (ordered by alphabet), as well as a validation set and a test set, each covering 20%. The dataset sizes are found at the bottom of Table 1. By putting all arguments from a debate into a single dataset, no specific topic knowledge can be gained from the training or validation set. We include all themes in all datasets, because we expect the set of themes to be stable.

We checked for duplicates. Among the 13 532 point and counters, 3407 appear twice, 723 three times, 36 four times, and 1 five times. We ensure that no true pair is used as a false pair in our tasks.

4 Counterargument Retrieval Tasks

Based on the new corpus, we define the following eight counterargument retrieval tasks of different complexity. All tasks consider all true argument-counterargument pairs, while differing in terms of what arguments (points and/or counters) from which context (same debate, same theme, or entire portal) are candidates for a given argument.

Same Debate: Opposing Counters All counters in the same debate with stance opposite to the given argument are candidates (Figure 2: a, b). The task is to find the best counterargument among all counters to the argument's stance.

Same Debate: Counters All counters in the same debate irrespective of their stance are candidates (Figure 2: a–c). The task is to find the best counterargument among all on-topic arguments phrased as counters.

		Training Set			Validation Set			Test Set		
Context	Candidate Counterarg's	True	False	Ratio	True	False	Ratio	True	False	Ratio
Same debate	Opposing counters	4 065	11 672	1:2.9	1 287	3 590	1:2.8	1 401	4 052	1:2.9
	Counters	4 065	27 024	1:6.6	1 287	8 348	1:6.5	1 401	9 312	1:6.6
	Opposing arguments	4 065	27 026	1:6.6	1 287	8 350	1:6.5	1 401	9 312	1:6.6
	Arguments	4 065	54 070	1:13.3	1 287	16 700	1:13.0	1 401	18 630	1:13.3
Same theme	Counters	4 065	1 616 000	1:398	1 287	176 266	1:137	1 401	189 870	1:136
	Arguments	4 065	3 232 038	1:795	1 287	352 536	1:274	1 401	379 746	1:271
Entire portal	Counters	4 065	16 517 994	1:4063	1 287	1 654 878	1:1286	1 401	1 961 182	1:1400
	Arguments	4 065	33 038 154	1:8127	1 287	3 309 760	1:2572	1 401	3 922 582	1:2800

Table 2: Number of true and false argument-counterargument pairs as well as their ratio for each evaluated context and type of candidate counterarguments in the three datasets. Each line defines one retrieval task.

Same Debate: Opposing Arguments All arguments in the same debate with opposite stance are candidates (Figure 2: a, b, d). The task is to find the best among all on-topic counterarguments.

Same Debate: Arguments All arguments in the same debate irrespective of their stance are candidates (Figure 2: a–e). The task is to find the best counterargument among all on-topic arguments.

Same Theme: Counters All counters from the same theme are candidates (Figure 2: a–c, f). The task is to find the best counterargument among all on-theme arguments phrased as counters.

Same Theme: Arguments All arguments from the same theme are candidates (Figure 2: a–g). The task is to find the best counterargument among all on-theme arguments.

Entire Portal: Counters All counters are candidates (Figure 2: a–c, f, h). The task is to find the best counterargument among all arguments phrased as counters.

Entire Portal: Arguments All arguments are candidates (Figure 2: a–i). The task is to find the best counterargument among all arguments.

Table 2 lists the numbers of true and false pairs for each task. Experiment files containing the file paths of all candidate pairs are provided in our corpus.

5 Retrieval of the Best Counterargument without Prior Topic Knowledge

The eight defined tasks indicate the subproblems of retrieving the best counterargument to a given argument: Finding all arguments that address the *same topic*, filtering those arguments with an *opposite stance* towards the topic, and identifying the *best counter* among these arguments. This section presents our approach to solving these problems computationally without prior knowledge of the argument's topic, based on the simultaneous similarity and dissimilarity of arguments.[2]

5.1 Topic as Word and Embedding Similarity

We do not reinvent the wheel to assess topical relevance, but rather follow common practice. Concretely, we hypothesize a candidate counterargument to be on-topic if it is similar to the argument in terms of its words and its embedding. We capture these two types of similarity as follows.

Word Argument Similarity To best represent the words in arguments, we did initial counterargument retrieval tests with token, stem, and lemma n-grams, $n \in \{1, 2, 3\}$. While the differences were not large, stems worked best and stem 1-grams sufficed. Both might be a consequence of the limited data size. In our experiments in Section 6, we determine the *stem 1-grams* to be considered on the training set of each task.

For word similarity computation, we tested four inverse vector-based distance measures: Cosine, Euclidean, Manhattan, and, Jaccard similarity (Cha, 2007). On the validation sets, the Manhattan similarity performed best, closely followed by the Jaccard similarity. Both clearly outperformed Euclidean and especially Cosine similarity. This suggests that the presence and absence of words are equally important and that outliers should not be punished more. For brevity, we report only results for the *Manhattan similarity* below.

[2] As indicated above, counters on idebate.org (including all true counterarguments) may also differ linguistically from points (all of which are false). However, we assume this to be a specific corpus bias and hence do not explicitly account for it. Section 6 will show whether having both points and counters as candidates makes counterargument retrieval harder.

Embedding Argument Similarity We evaluated five pretrained word embedding models for representing arguments in first tests: GoogleNews-vectors (Mikolov et al., 2013), ConceptNet Numberbatch (Speer et al., 2017), wiki-news-300d-1M, wiki-news-300d-1M-subword, and crawl-300d-2M (Mikolov et al., 2017). The former two were competitive, the others performed notably worse. Since *ConceptNet Numberbatch* is smaller and supposed to have less bias, we used it in all experiments.

To capture argument-level embedding similarity, we compared the four inverse vector-based distance measures above on average word embeddings against the inverse Word Mover's distance, which quantifies the optimum alignment of two word embedding sequences (Kusner et al., 2015). This *Word Mover's similarity* consistently beat the others, so we decided to restrict our view to it.

5.2 Stance as Topic Dissimilarity

Stance classification without prior topic knowledge is challenging: While we can compare the topics of any two arguments, it is impossible in general to infer the stance of the specific aspects invoked by one argument to those of the other. As sketched in Section 2, related work employs external knowledge to infer stance relations of aspects and topics (Bar-Haim et al., 2017) or trains classifying attack relations (Cabrio and Villata, 2012). Unfortunately, both does not apply to topics unseen before.

For argument pairs invoking similar aspects, a way to go is in principle to assess sentiment polarity; intuitively, two arguments with the same topic but opposite sentiment have opposing stance. However, we tested topic-agnostic sentiment lexicons (Baccianella et al., 2010) and state-of-the-art sentiment classifiers, trained on large-scale multiple-domain review data (Prettenhofer and Stein, 2010; Joulin et al., 2017). The correlation between sentiment and stance differences of training arguments was close to zero. A possible explanation is the limited explicitness of sentiment on idebate.org, making the lexicons and classifiers fail there.

Other related work suggests that the vocabulary of opposing sides differs (Beigman Klebanov et al., 2010). We thus checked on the training set whether counterarguments are similar in their embeddings but dissimilar in their words. The measures above did not support this hypothesis, i.e., both embedding and word similarity increased the likelihood of a candidate counterargument being the best. Still, there must be a difference between an argument and its counterargument by concept. As a solution, we capture dissimilarity with the same similarity functions as above, but we change the granularity level on which we measure similarity.

5.3 Simultaneous Similarity and Dissimilarity

The arising question is how to assess similarity and dissimilarity at the same time. We hypothesize the best counterargument to be very similar in overall terms, but very dissimilar in certain respects. To capture this intuition, we rely on expert knowledge from argumentation theory (see Section 2).

Word and Embedding Unit Similarities In particular, we follow the notion that a counterargument attacks either the conclusion of an argument, the argument's premises, or both. As a consequence, we compute two word and two embedding similarities as specified above for each candidate counterargument; once to the argument's conclusion (called w_c and e_c for words and embeddings respectively) and once to the argument's premises (w_p and e_p).

Now, to capture similarity and dissimilarity simultaneously, we need multiple ways to aggregate conclusion and premise similarities. As we do not generally know which argument unit is attacked, we resort to four standard aggregation functions that generalize over the unit similarities. For words, these are the following *word unit similarities*:

$$\begin{aligned} w_{\downarrow} &:= \min\{w_c, w_p\} & \qquad w_{\times} &:= w_c \cdot w_p \\ w_{\uparrow} &:= \max\{w_c, w_p\} & \qquad w_{+} &:= w_c + w_p \end{aligned}$$

Accordingly, we define four respective *embedding unit similarities*, $e_{\downarrow}$, $e_{\uparrow}$, $e_{\times}$, and e_{+}.

As mentioned above, both word similarity and embedding similarity positively affect the likelihood that a candidate is the best counterargument. Therefore, we combine each pair of similarities as $w_{\downarrow} + e_{\downarrow}$, $w_{\uparrow} + e_{\uparrow}$, $w_{\times} + e_{\times}$, and $w_{+} + e_{+}$, but we also evaluate their impact in isolation below.[3]

Counterargument Scoring Model Based on the unit similarities, we finally define a scoring model for a given pair of argument and candidate counterargument. The model includes two unit similarity values, sim and $dissim$, but $dissim$ is subtracted from sim, such that it actually favors dissimilarity. Thereby, we realize the topic and

[3] In principle, other unit similarities could be used for words than for embeddings. However, we decided to couple them to maintain interpretability of our experiment results.

stance similarity sketched in Figure 1. We weight the two values with a damping factor α:

$$\alpha \cdot sim \; - \; (1-\alpha) \cdot dissim$$

where $sim, dissim \in \{w_\downarrow + e_\downarrow, w_\uparrow + e_\uparrow, w_\times + e_\times, w_+ + e_+\}$ and $sim \neq dissim$.

The general idea of the scoring model is that sim rewards one type of similarity, whereas subtracting $dissim$ punishes another type. We seek to thereby find the most dissimilar candidate among the similar candidates. The model is meant to give a higher score to a pair the more likely the candidate is the best counterargument to the argument, so the scores can be used for ranking.

What combination of sim and $dissim$ turns out best, is hard to foresee and may depend on the retrieval task at hand. We hence evaluate different combinations empirically below. The same holds for the damping factor $\alpha \in [0, 1]$. If our hypothesis on similarity and dissimilarity is true, then the best α should be close to but lower than 1. Conversely, if $\alpha = 1.0$ achieves the best performance, then only similarity would be captured by our model.

6 Evaluation

We now report on systematic ranking experiments with our counterargument scoring model. The goal is to evaluate on all eight retrieval tasks from Section 4 to what extent our hypothesis holds that the best counterargument to an argument invokes the same aspects while having opposing stance. The Java source code of the experiments is available at: http://www.arguana.com/software

6.1 Experimental Set-up

We evaluated the following set-up of tasks, data, measures, baselines, and approaches.

Tasks We tackled each of the eight retrieval tasks as a ranking problem, i.e., we aimed to rank the best counterargument to each argument highest, given all candidates. Accordingly, only one candidate counterargument per argument is correct.[4]

Data Table 2 has shown the true and false argument pairs in all datasets. We undersampled each training set, resulting in 4065 true and 4065 false training pairs in all tasks.[5] Our model does not do any learning-to-rank on these pairs, but we derived lexicons for the word similarities from them (all stems included in at least 1% of all pairs). As detailed below, we then determined the best model configurations on the validation sets and evaluated these configurations on the test sets.

Measures As only one candidate is true per argument, we report the *accuracy@1* of each approach, i.e., the percentage of arguments for which the true counterargument was ranked highest. Besides, we compute the rounded *mean rank* of the best counterargument in all rankings, reflecting the average performance of an approach. Exemplarily, we also mention the *mean reciprocal rank (MRR)*, which is more sensitive to outliers.

Baselines A trivial way to address the given tasks is to pick any candidate by chance for each argument. This *random baseline* allows quantifying the impact of other approaches. As counterargument retrieval has not been tackled yet, we do not use any existing baseline.[6] Instead, we evaluate the effects of the different building blocks of our scoring model. On one hand, we check the need for distinguishing conclusions and premises by comparing to the word argument similarity (w) and the embedding argument similarity (e). On the other hand, we consider all eight word and embedding unit similarities ($w_\downarrow$, $w_\uparrow$, ..., e_+) as baselines, in order to see whether and how to best aggregate them.

Approaches After initial tests, we reduced the set of tested values of the damping factor α in our scoring model to $\{0.8, 0.9, 1.0\}$. On the validation sets of the first six tasks,[7] we then analyzed all possible combinations of $w_\downarrow + e_\downarrow$, $w_\uparrow + e_\uparrow$, $w_\times + e_\times$, $w_+ + e_+$, as well as $w + e$ for sim and $dissim$. Three configurations of the model turned out best:

$$\begin{aligned} we \quad &:= \quad 1.0 \cdot (w_\times + e_\times) \\ we_\downarrow \quad &:= \quad 0.9 \cdot (w_\times + e_\times) - 0.1 \cdot (w_\downarrow + e_\downarrow) \\ we_\uparrow \quad &:= \quad 0.9 \cdot (w_+ + e_+) - 0.1 \cdot (w_\uparrow + e_\uparrow) \end{aligned}$$

[4] One alternative would be to see each argument pair as one instance of a classification problem. However, our preliminary tests confirmed the intuition that identifying the best counterargument is hard without knowing the other candidates, i.e., there is no general (dis)similarity threshold that makes an argument the best counterargument. Rather, how similar or dissimilar a counterargument needs to be depends on the topic and on the other candidates. Another alternative would be to treat all candidates for an argument as one instance, but this makes the experimental set-up very intricated.

[5] Undersampling was done stratified, such that the same number of false counterarguments was taken from each type, b–i, in Figure 2 that is relevant in the respective task.

[6] Notice, though, that we tested a number of approaches to identify opposing stance, as discussed in Section 5.

[7] We did not expect "game-changing" validation set results for the last two tasks and, so, left them out for time reasons.

		Same Debate								**Same Theme**				**Entire Portal**			
		Opp. Ctr.'s		*Counters*		*Opposing*		*Arguments*		*Counters*		*Arguments*		*Counters*		*Arguments*	
#	**Baseline / Approach**	**@1**	**R**	**@1**	**R**	**@1**	**R**	**@1**	**R**	**@1**	**R**	**@1**	**R**	**@1**	**R**	**@1**	**R**
w	Word argument similarity	65.9	2	48.5	**2**	42.5	3	30.0	4	44.1	5	28.3	10	39.7	22	21.8	49
e	Embedding argument similarity	62.9	2	44.6	**2**	51.6	**2**	36.8	4	38.8	7	32.9	10	34.2	39	23.9	55
$w_{\downarrow}$	Word unit similarity minimum	53.8	2	38.4	3	45.9	3	33.7	5	28.5	22	24.8	42	21.4	206	18.5	403
$w_{\uparrow}$	Word unit similarity maximum	66.1	2	48.0	**2**	44.0	3	30.2	4	44.0	5	28.3	9	38.0	21	21.2	44
$w_{\times}$	Word unit similarity product	64.9	2	49.5	3	56.1	**2**	40.7	4	44.3	18	36.8	35	37.8	177	26.8	354
w_{+}	Word unit similarity sum	71.5	**1**	53.7	**2**	54.1	**2**	39.1	4	49.0	4	36.8	7	44.7	17	28.6	33
$e_{\downarrow}$	Embedding unit sim. minimum	61.6	2	44.9	3	43.4	3	32.1	4	37.8	7	27.4	13	32.5	42	20.7	74
$e_{\uparrow}$	Embedding unit sim. maximum	63.4	2	44.5	**2**	47.5	**2**	33.2	4	39.8	5	29.8	8	32.1	20	20.1	33
$e_{\times}$	Embedding unit sim. product	69.7	**1**	52.0	**2**	55.4	**2**	41.0	**3**	44.3	4	37.1	6	43.2	14	27.8	21
e_{+}	Embedding unit sim. sum	69.7	**1**	51.8	**2**	55.4	**2**	40.5	**3**	47.5	4	36.8	6	43.0	13	27.6	21
we	$1.0 \cdot (w_{\times} + e_{\times})$	72.1	**1**	55.2	**2**	‡**60.3**	**2**	†**44.9**	**3**	50.4	4	40.9	7	46.0	19	32.2	34
$we_{\downarrow}$	$0.9 \cdot (w_{\times} + e_{\times}) - 0.1 \cdot (w_{\downarrow} + e_{\downarrow})$	72.0	**1**	55.5	**2**	59.5	**2**	44.1	**3**	51.3	4	†**41.0**	7	46.3	19	31.7	35
$we_{\uparrow}$	$0.9 \cdot (w_{+} + e_{+}) - 0.1 \cdot (w_{\uparrow} + e_{\uparrow})$	†**74.5**	**1**	†**57.7**	**2**	59.6	**2**	44.1	**3**	‡**54.2**	**3**	40.8	**5**	‡**50.0**	**9**	‡**32.4**	**15**
r	Random baseline	25.7	2	13.1	4	13.1	4	7.0	7	0.7	69	0.4	137	0.1	701	0.0	1401

Table 3: Test set accuracy of ranking the best counterargument highest (**@1**) and mean rank (**R**) for 14 baselines and approaches (w, e, $w_{\downarrow}$, ..., r) in all eight tasks (given by **Context** and *Candidates*). Each best accuracy value (bold) significantly outperforms the best baseline with 99% (†) or 99.9% (‡) confidence.

we was best on the validation set of *Same Debate: Opposing Arguments* (accuracy@1: 62.1) and $we_{\downarrow}$ on the one of *Same Debate: Arguments* (49.0). All other tasks were dominated by $we_{\uparrow}$. Especially, $we_{\uparrow}$ was better than $1.0 \cdot (w_{+} + e_{+})$ in all of them with clear leads of up to 2.2 points. This underlines the importance of modeling dissimilarity for counterargument retrieval. We took we, $we_{\downarrow}$, and $we_{\uparrow}$ as our approaches for the test set.[8]

6.2 Results

Table 3 shows the accuracy@1 and the mean rank of all baselines and approaches on each of the eight given retrieval tasks.

Overall, the counter-only tasks seem slightly harder within the *same debate* (comparing *Counters* to *Opposing*), i.e., stance is harder to assess than topical relevance. Conversely, the other *Counters* tasks seem easier, suggesting that topically close but false candidate counterarguments with the same stance as the argument (which are not included in any *Counters* task) are classified wrongly most often. Besides, these results support that potential differences in the phrasing of counters are not exploited, as desired.

The accuracy of the standard similarity measures, w and e, goes from 65.9 and 62.9 respectively in the smallest task down to 21.8 and 23.9 in the largest. w is stronger when only counters are candidates, e otherwise. This implies that words capture differences between the best and other counters, whereas embeddings rather help discard false candidates with the same stance as the argument.

From the eight unit similarity baselines, w_{+} performs best on five tasks ($e_{\times}$ twice, $w_{\times}$ once). w_{+} finds 71.5% true counterarguments among all opposing counters in a debate, and 28.6% among all test arguments from the entire portal. In that task, however, the mean ranks of w_{+} (33) and particularly of $w_{\times}$ (354) are much worse than for $e_{\times}$ (21), meaning that words are insufficient to robustly find counterarguments.

we, $we_{\downarrow}$, and $we_{\uparrow}$ outperform all baselines in all tasks, improving the accuracy by 8.1 (*Same Theme: Arguments*) to 10.3 points (*Entire Portal: Counters*) over w and e, and at least 3.0 over the best baseline in each task. Among all opposing arguments from the same debate (true-to-false ratio 1:6.6), we finds 60.3% of the best counterarguments, 44.9% when all arguments are given (1:13.3).

The winner in our evaluation is $we_{\uparrow}$, though, being best in five of the eight tasks. It found the true among all opposing counters in 74.5% of all cases, and about every third time (32.4) among all 2801 test set arguments; a setting where the random baseline has virtually no chance. Given all arguments from the same theme, $we_{\uparrow}$ puts the best counterargument at a mean rank of 5 (MRR 0.58), and for the entire portal still at 15 (MRR 0.5).

[8] All validation set results are found in the supplementary material, which we provide at http://www.arguana.com/publications

Entire Portal: Arguments		Accuracy@1		Mean Rank	
Theme	**Arguments**	w_+	$we_\uparrow$	w_+	$we_\uparrow$
Culture	69	31.9	36.2	12	9
Digital freedoms	61	37.7	44.3	58	20
Economy	125	27.2	25.6	21	10
Education	81	38.3	39.5	36	17
Environment	46	17.4	21.7	22	7
Free speech debate	58	10.3	12.1	130	55
Health	77	28.6	36.4	26	14
International	271	25.8	31.4	31	19
Law	134	38.8	38.1	16	8
Philosophy	85	34.1	38.8	29	14
Politics	202	28.7	33.2	28	11
Religion	45	24.4	33.3	58	8
Science	57	19.3	28.1	6	5
Society	60	16.7	20.0	45	22
Sport	30	43.3	46.7	35	9
All themes	**1401**	**28.6**	**32.4**	**33**	**15**

Table 4: Accuracy@1 and mean rank of the best baseline (w_+) and approach ($we_\uparrow$) on each theme when all 2801 test set arguments are candidates.

Although our scoring model thus does not *solve* the retrieval tasks, we conclude that it serves as a robust approach to rank the best counterargument high.

To test significance, we separately computed the accuracy@1 for the arguments from each theme. The differences between the 15 values of the best approach on each task and those of the best baseline (w_+, $w_\times$, or $e_\times$) were normally distributed. Since the baselines and approaches are dependent, we used a one-tailed dependent t-test with paired samples. As Table 3 specifies, our approaches are consistently better, partly with at least 99% confidence, partly even with 99.9% confidence.

In Table 4, we exemplarily detail the comparison of the best approach ($we_\uparrow$) to the best baseline (w_+) on *Entire Portal: Arguments*. The mean ranks across themes underline the robustness of $we_\uparrow$, being in the top 10 for 7 and in the top 20 even for 13 themes. Still, the accuracy@1 of both w_+ and $we_\uparrow$ varies notably, in case of $we_\uparrow$ from 12.1 for *free speech debate* to 46.7 for *sport*. For free speech debates (e.g., "This House would criminalise blasphemy"), we observed that their arguments tend to be overproportionally long, which might lead to deviating similarities. In case of sports, the topical specificity (e.g., "This House would ban boxing") reduces the probability of mistakenly choosing candidates from other themes.

Free speech debate turned out the hardest theme in seven tasks, *health* in the remaining one. Besides sports, in some tasks the best results were obtained for *religion* and *science*, both of which share the characteristic of dealing with very specific topics.[9]

7 Conclusion

This paper has asked how to find the best counterargument to any argument without prior knowledge of the argument's topic. We did *not* aim to engineer the best approach to this retrieval task, but to study whether we can model the simultaneous similarity and dissimilarity of a counterargument to an argument computationally. For the restricted domain of debate portal arguments, our main result is quite intriguing: The best model ($we_\uparrow$) rewards a high overall similarity to the argument's conclusion and premises while punishing a too high similarity to either of them. Despite its simplicity, $we_\uparrow$ found the best counterargument among 2801 candidates in almost a third of all cases, and ranked it into the top 15 on average. This speaks for our hypothesis that the best counterargument often just addresses the same topical aspects with opposite stance.

Of course, our hypothesis is simplifying, i.e., there are counterarguments that will not be found based on aspect and stance similarity only. Apart from some hyperparameters, however, our model is unsupervised and it does not make any assumption about an argument's topic. Hence, it applies to any argument, given a pool of candidate counterarguments. While the model can be considered *open-topic*, a next step will be to study counterargument retrieval *open-source*.

We are confident that the modeled intuition generalizes beyond idebate.org. To obtain further insights into the nature of counterarguments, deeper linguistic analysis along with supervised learning may be needed, though. We provide a corpus to train respective approaches, but leave the according research to future work.

The intended practical application of our model is to retrieve counterarguments in automatic debating technologies (Rinott et al., 2015) and argument search (Wachsmuth et al., 2017b). While debate portal arguments are often suitable in this regard, in general not always a real counterargument exists to an argument. Still, returning one that addresses similar aspects with opposite stance makes sense then. An alternative would be to *generate* counterarguments, but we believe that humans are better than machines in writing them — currently.

[9] The individual results of the best approach and baseline on each theme are also found in the supplementary material.

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*, pages 118–128. 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*, pages 1395–1404. Association for Computational Linguistics.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*. European Languages Resources Association (ELRA).

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*, pages 251–261. Association for Computational Linguistics.

Beata Beigman Klebanov, Eyal Beigman, and Daniel Diermeier. 2010. Vocabulary choice as an indicator of perspective. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 253–257. Association for Computational Linguistics.

Filip Boltužić and Jan Šnajder. 2014. Back up your stance: Recognizing arguments in online discussions. In *Proceedings of the First Workshop on Argumentation Mining*, pages 49–58. Association for Computational Linguistics.

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.

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)*, pages 208–212. Association for Computational Linguistics.

Sung-Hyuk Cha. 2007. Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1(4):300–307.

Oana Cocarascu and Francesca Toni. 2017. Identifying attack and support argumentative relations using deep learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1374–1379. Association for Computational Linguistics.

T. Edward Damer. 2009. *Attacking Faulty Reasoning: A Practical Guide to Fallacy-Free Arguments*, 6th edition. Wadsworth, Cengage Learning, Belmont, CA.

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*, pages 987–996. Association for Computational Linguistics.

Trudy Govier. 2010. *A Practical Study of Argument*, 7th edition. Wadsworth, Cengage Learning, Belmont, CA.

Ivan Habernal and Iryna Gurevych. 2016. What makes a convincing argument? Empirical analysis and detecting attributes of convincingness in web argumentation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1214–1223. Association for Computational Linguistics.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. Before name-calling: Dynamics and triggers of ad hominem fallacies in web argumentation. In *16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, to appear.

Yufang Hou and Charles Jochim. 2017. Argument relation classification using a joint inference model. In *Proceedings of the 4th Workshop on Argument Mining*, pages 60–66. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From word embeddings to document distances. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, pages 957–966.

John Lawrence and Chris Reed. 2017. Mining argumentative structure from natural language text using automatically generated premise-conclusion topic models. In *Proceedings of the 4th Workshop on Argument Mining*, pages 39–48. Association for Computational Linguistics.

Stefano Menini, Federico Nanni, Simone Paolo Ponzetto, and Sara Tonelli. 2017. Topic-based agreement and disagreement in us electoral manifestos. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2938–2944. Association for Computational Linguistics.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2017. Advances in pre-training distributed word representations. *CoRR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, pages 3111–3119.

Vlad Niculae, Joonsuk Park, and Claire Cardie. 2017. Argument mining with structured SVMs and RNNs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 985–995. Association for Computational Linguistics.

Andreas Peldszus and Manfred Stede. 2015. Towards detecting counter-considerations in text. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 104–109. Association for Computational Linguistics.

Peter Prettenhofer and Benno Stein. 2010. Cross-language text classification using structural correspondence learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1118–1127. Association for Computational Linguistics.

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*, pages 440–450. Association for Computational Linguistics.

Haggai Roitman, Shay Hummel, Ella Rabinovich, Benjamin Sznajder, Noam Slonim, and Ehud Aharoni. 2016. On the retrieval of wikipedia articles containing claims on controversial topics. In *Proceedings of the 25th International Conference on World Wide Web, Companion Volume*, pages 991–996.

Parinaz Sobhani, Diana Inkpen, and Stan Matwin. 2015. From argumentation mining to stance classification. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 67–77. Association for Computational Linguistics.

Robert Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4444–4451.

Christian Stab and Iryna Gurevych. 2016. Recognizing the absence of opposing arguments in persuasive essays. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 113–118. Association for Computational Linguistics.

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2016. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th International World Wide Web Conference*, pages 613–624.

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

Henning Wachsmuth, Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Tim Alberdingk 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*, pages 176–187. Association for Computational Linguistics.

Henning Wachsmuth, Martin Potthast, Khalid Al Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. 2017b. Building an argument search engine for the web. In *Proceedings of the 4th Workshop on Argument Mining*, pages 49–59. Association for Computational Linguistics.

Henning Wachsmuth, Benno Stein, and Yamen Ajjour. 2017c. "PageRank" for Argument Relevance. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1117–1127. Association for Computational Linguistics.

Douglas Walton. 2006. *Fundamentals of Critical Argumentation*. Cambridge University Press.

Douglas Walton. 2009. Objections, rebuttals and refutations. pages 1–10.

Justine Zhang, Ravi Kumar, Sujith Ravi, and Cristian Danescu-Niculescu-Mizil. 2016. Conversational flow in Oxford-style debates. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 136–141. Association for Computational Linguistics.