Classification and Clustering of Arguments with Contextualized Word Embeddings

Nils Reimers, Benjamin Schiller, Tilman Beck,
Johannes Daxenberger, Christian Stab, Iryna Gurevych
Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technische Universität Darmstadt
www.ukp.tu-darmstadt.de

Abstract

We experiment with two recent contextualized word embedding methods (ELMo and BERT) in the context of open-domain argument search. For the first time, we show how to leverage the power of contextualized word embeddings to classify and cluster topic-dependent arguments, achieving impressive results on both tasks and across multiple datasets. For argument classification, we improve the state-of-the-art for the UKP Sentential Argument Mining Corpus by 20.8 percentage points and for the IBM Debater - Evidence Sentences dataset by 7.4 percentage points. For the understudied task of argument clustering, we propose a pre-training step which improves by 7.8 percentage points over strong baselines on a novel dataset, and by 12.3 percentage points for the Argument Facet Similarity (AFS) Corpus.[1]

1 Introduction

Argument mining methods have been applied to different tasks such as identifying reasoning structures (Stab and Gurevych, 2014), assessing the quality of arguments (Wachsmuth et al., 2017), or linking arguments from different documents (Cabrio and Villata, 2012). Broadly speaking, existing methods either approach argument mining from the discourse-level perspective (aiming to analyze local argumentation structures), or from an information-seeking perspective (aiming to detect arguments relevant to a predefined topic). While discourse-level approaches mostly focus on the analysis of single documents or document collections (Eger et al., 2017), information-seeking approaches need to be capable of dealing with heterogeneous sources and topics (Shnarch et al., 2018) and also face the problem of redundancy, as

arguments might be repeated across sources. As a result, this perspective naturally calls for a subsequent clustering step, which is able to identify and aggregate similar arguments for the same topic. In this work, we focus on the latter perspective, referring to it as *open-domain* argument search, and show how contextualized word embeddings can be leveraged to overcome some of the challenges involved in topic-dependent argument classification and clustering.

Identifying arguments for unseen topics is a challenging task for machine learning systems. The lexical appearance for two topics, e.g. "net neutrality" and "school uniforms", is vastly different. Hence, in order to perform well, systems must develop a deep semantic understanding of both the topic as well as the sources to search for arguments. Even more so, clustering similar arguments is a demanding task, as fine-grained semantic nuances may determine whether two arguments (talking about the same topic) are similar. Figure 1 gives an example of arguments on the topic "net neutrality". Both arguments center around the aspect of "equal access for every Internet user" but are differently phrased.

A1 The ultimate goal is fast, affordable, open Internet access for everyone, everywhere.
A2 If this does not happen, we will create an Internet where only users able to pay for privileged access enjoy the network's full capabilities.

Figure 1: Similar *pro* arguments for the topic "net neutrality".

Contextualized word embeddings, especially ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018) could offer a viable solution to this problem. In contrast to traditional word embeddings like word2vec (Mikolov et al., 2013) or

---

GloVe (Pennington et al., 2014), these methods compute the embeddings for a sentence on the fly by taking the context of a target word into account. This yields word representations that better match the specific sense of the word in a sentence. In cross-topic scenarios, with which we are dealing in open-domain argument search, contextualized representations need to be able to adapt to new, unseen textual topics. We thus analyze ELMo and BERT in a cross-topic scenario for the tasks of argument classification and clustering on four different datasets. For argument classification, we use the *UKP Sentential Argument Mining Corpus* by Stab et al. (2018b) and the *IBM Debater®: Evidence Sentences* corpus by Shnarch et al. (2018). For argument clustering, we introduce a novel corpus on *aspect-based argument clustering* and evaluate the proposed methods on this corpus as well as on the *Argument Facet Similarity Corpus* (Misra et al., 2016).

The contributions in this publications are: (1) We frame the problem of open-domain argument search as a combination of topic-dependent argument classification and clustering and discuss how contextualized word embeddings can help to improve these tasks across four different datasets. (2) We show that our suggested methods improve the state-of-the-art for argument classification when fine-tuning the models, thus significantly reducing the gap to human performance. (3) We introduce a novel corpus on aspect-based argument similarity and demonstrate how contextualized word embeddings help to improve clustering similar arguments in a supervised fashion with little training data.

We present the four different datasets used in this work in Section 3, before we discuss our experiments and results on argument classification and clustering in Sections 4 and 5. We conclude our findings for open-domain argument search in Section 6.

2 Related Work

In the following, we concentrate on the fundamental tasks involved in open-domain argument search. First, we discuss work that experiments with sentence-level argument classification. Second, we review work that provides us with the necessary tools to cluster extracted arguments by their similarity. Third, we take a deeper look into contextualized word embeddings.

Argument Classification, as viewed in this work, aims to identify topic-related, sentence-level arguments from (heterogeneous) documents. Levy et al. (2014) identify context-dependent claims (CDCs) by splitting the problem into smaller sub-problems. Rinott et al. (2015) extend this work with a pipeline of feature-based models that find and rank supporting evidence from Wikipedia for the CDCs. However, neither of these approaches leverage the potential of word embeddings in capturing semantic relations between words.

Shnarch et al. (2018) aim to identify topic-dependent evidence sentences by blending large automatically generated training sets with manually annotated data as initialization step. They use a BiLSTM with GloVe embeddings and integrate the topic via attention. For topic-dependent argument detection, Stab et al. (2018b) deploy a modified LSTM-cell that is able to directly integrate topic information. They show the importance of topic information by outperforming a BiLSTM baseline by around 4.5pp. Yet, their best model only shows mediocre recall for arguments, while showing an even lower precision when compared to their baseline. As argument classification is the first logical step in open-domain argument search, a low performance would eventually propagate further down to the clustering of similar arguments. Hence, in this work, we aim to tackle this problem by leveraging superior contextualized language models to improve on precision and recall of argumentative sentences.

Argument Clustering aims to identify similar arguments. Previous research in this area mainly used feature-based approaches in combination with traditional word embeddings like word2vec or GloVe. Boltužić and Šnajder (2015) applied hierarchical clustering on semantic similarities between users' posts from a two-side online debate forum using word2vec. Wachsmuth et al. (2018) experimented with different word embeddings techniques for (counter)argument similarity. Misra et al. (2016) presented a new corpus on argument similarity on three topics. They trained a Support Vector Regression model using different hand-engineered features including custom trained word2vec. Trabelsi and Zaïane (2015) used an augmented LDA to automatically extract coherent words and phrases describing arguing expressions and apply constrained cluster-

ing to group similar viewpoints of topics.

In contrast to previous work, we apply argument clustering on a dataset containing both relevant and non-relevant arguments for a large number of different topics which is closer to a more realistic setup.

Contextualized word embeddings compute a representation for a target word based on the specific context the word is used within a sentence. In contrast, traditional word embedding methods, like word2vec or GloVe, words are always mapped to the same vector. Contextualized word embeddings tackle the issue that words can have different senses based on the context. Two approaches that became especially popular are ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018).

ELMo (Embeddings from Language Models) representations are derived from a bidirectional language model, that is trained on a large corpus. Peters et al. combine a character-based CNN with two bidirectional LSTM layers. The ELMo representation is then derived from all three layers.

BERT (Bidirectional Encoder Representations from Transformers) uses a deep transformer network (Vaswani et al., 2017) with 12 or 24 layers to derive word representations. Devlin et al. presented two new pre-training objectives: the "masked language model" and the "next sentence prediction" objectives. They demonstrate that the pre-trained BERT models can be fine-tuned for various tasks, including sentence classification and sentence-pair classification.

ELMo and BERT were primarily evaluated on datasets where the test and training sets have comparable distributions. In cross-topic setups, however, the distributions for training and testing are vastly different. It is unclear, whether ELMo and BERT will be able to adapt to this additional challenge for cross-topic argument mining.

3 Datasets

No dataset is available that allows evaluating open-domain argument search end-to-end. Hence, we analyze and evaluate the involved steps (argument classification and clustering) independently.

3.1 Argument Classification

To our knowledge, to date there are only two suitable corpora for the task of topic-dependent argument classification.

UKP Corpus. The *UKP Sentential Argument Mining Corpus* by Stab et al. (2018b) (henceforth: *UKP corpus*) annotated 400 documents with 25,492 sentences on eight controversial topics with the labels: *pro/con/no argument*.

IBM Corpus. The *IBM Debater®: Evidence Sentences* by Shnarch et al. (2018) (henceforth: *IBM corpus*) contains 118 topics drawn from different debate portals. For each topic, Shnarch et al. (2018) extracted sentences from Wikidata that were in turn annotated by crowd-workers (10 for each topic-sentence pair) with one of the two labels: *evidence* or *no evidence* in regard to the topic.

3.2 Argument Clustering

Topic-dependent argument clustering is an understudied problem with few resources available. Arguments on controversial topics usually address a limited set of aspects, for example, many arguments on "nuclear energy" address safety concerns. Argument pairs addressing the same aspect should be assigned a high similarity score, and arguments on different aspects a low score. To date, the only available resource of that kind we are aware of, is the Argument Facet Similarity (AFS) Corpus (Misra et al., 2016).

AFS Corpus. The AFS corpus annotates similarities of arguments pairwise. Misra et al. (2016) aimed to create automatic summaries for controversial topics. As an intermediate step, they extracted 6,000 sentential argument pairs from curated online debating platforms for three topics and annotated them on a scale from 0 ("different topic") to 5 ("completely equivalent"). A drawback of this corpus is that the arguments are curated, i.e., the dataset does not include noise or non-relevant arguments. Furthermore, the corpus covers only three different topics.

UKP ASPECT Corpus. To remedy these shortcomings, we create a new corpus with annotations on similar and dissimilar sentence-level arguments (Stab et al., 2018b), referred to as the Argument Aspect Similarity (UKP ASPECT) Corpus in the following.[2] The UKP ASPECT corpus consists of sentences which have been identified as arguments for given topics using the ArgumenText system (Stab et al., 2018a). The ArgumenText system expects as input an arbitrary topic (query) and searches a large web crawl for relevant docu-

---

[2]The dataset is available at http://www.ukp.tu-darmstadt.de/data

ments. Finally, it classifies all sentences contained in the most relevant documents for a given query into *pro*, *con* or non-arguments (with regard to the given topic).

We picked 28 topics related to currently discussed issues from technology and society. To balance the selection of argument pairs with regard to their similarity, we applied a weak supervision approach. For each of our 28 topics, we applied a sampling strategy that picks randomly two *pro* or *con* argument sentences at random, calculates their similarity using the system by Misra et al. (2016), and keeps pairs with a probability aiming to balance diversity across the entire similarity scale. This was repeated until we reached 3,595 arguments pairs, about 130 pairs for each topic.

The argument pairs were annotated on a range of three degrees of similarity (no, some, and high similarity) with the help of crowd workers on the Amazon Mechanical Turk platform. To account for unrelated pairs due to the sampling process, crowd workers could choose a fourth option.[3] We collected seven assignments per pair and used Multi-Annotator Competence Estimation (MACE) with a threshold of 1.0 (Hovy et al., 2013) to consolidate votes into a gold standard. About 48% of the gold standard pairs are labeled with *no similarity*, whereas about 23% resp. 13% are labeled with *some* resp. *high similarity*. Furthermore, 16% of the pairs were labeled as containing invalid argument(s) (e.g. irrelevant to the topic at hand).

We asked six experts (graduate research staff familiar with argument mining) to annotate a random subset of 50 pairs from 10 topics. The resulting agreement among experts was Krippendorff's $\alpha = 0.43$ (binary distance) resp. 0.47 (weighted distance[4]), reflecting the high difficulty of the task. Krippendorff's α agreement between experts and the gold standard from crowd workers was determined as 0.54 (binary) resp. 0.55 (weighted distance).

4 Argument Classification

As a first task in our pipeline of open-domain argument search, we focus on topic-dependent, sentence-level argument classification. To prevent

the propagation of errors to the subsequent task of argument clustering, it is paramount to reach a high performance in this step.

4.1 Experimental Setup

For the UKP Corpus, we use the proposed evaluation scheme by Stab et al. (2018b): The models are trained on the train split (70% of the data) of seven topics, tuned on the dev split (10%) of these seven topics, and then evaluated on the test split (20%) of the eighth topic. A macro F_1-score is computed for the 3-label classes and scores are averaged over all topics and over ten random seeds. For the IBM Corpus, we use the setup by Shnarch et al. (2018): Training on 83 topics (4,066 sentences) and testing on 35 topics (1,719 sentences). We train for five different random seeds and report the average accuracy over all runs.

4.2 Methods

We experiment with a number of different models and distinguish between models which use topic information and ones that do not.

bilstm. This model was presented as a baseline by Stab et al. (2018b). It trains a bi-directional LSTM network on the sentence, followed by a softmax classifier and has no information about the topic. As input, pre-trained word2vec embeddings (Google News dataset) were used.

biclstm. Stab et al. (2018b) presented the contextualized LSTM (clstm), which adds topic information to the *i*- and *c*-cells of the LSTM. The topic information is represented by using pre-trained word2vec embeddings.

IBM. Shnarch et al. (2018) blend large automatically generated training sets with manually annotated data in the initialization step. They use an LSTM with 300-d GloVe embeddings and integrate the topic via attention. We re-implemented their system, as no official code is available.

We experiment with these three models by replacing the word2vec / GloVe embeddings with ELMo and BERT embeddings. The ELMo embeddings are obtained by averaging the output of the three layers from the pre-trained 5.5B ELMo model. For each token in a sentence, we generate a BERT embedding with the pre-trained BERT-large-uncased model.

Further, we evaluate fine-tuning the transformer network from BERT for our datasets:

BERT. We add a softmax layer to the output of the first token from BERT and fine-tune the net-

---

[3]The exact layout of the Human Intelligence Task (HIT) guidelines, as well as agreement statistics can be seen in the appendix.

[4]Reduced distance of 0.5 between *high* and *some similarity*, otherwise 1.

work for three epochs with a batch size of 16 and a learning rate of 2e-5. We only present the sentence to the BERT model.

BERT$_{topic}$. We add topic information to the BERT network by changing the input to the network. We concatenate the topic and the sentence (separated by a special `[SEP]`-token) and fine-tune the network as mentioned before.

4.3 Results and Analysis

In the following, we present and analyze the results.

UKP Corpus. Replacing traditional embeddings in the *bilstm* by contextualized word embeddings improves the model's performance by around 6pp and 8pp in F_1 for ELMo and BERT (see Table 1). The fine-tuned *BERT-large* improves by even 12pp over the baseline *bilstm* and by this also outperforms *bilstm$_{BERT}$* by around 4pp. Hence, using an intermediary BiLSTM layer for the *BERT* model even hurts the performance.

Using ELMo and BERT embeddings in the topic-integrating *biclstm* model significantly decreases the performance, as compared to their performance in the *bilstm*. The contextualized word embedding for a topic is different to the one of a topic appearing in a sentence and the *biclstm* fails to learn a connection between them.

Including the topic into the fine-tuned BERT models increases the F_1 score by approx. 14.5pp and 13pp for *BERT-base* and *BERT-large*. This is due to a vast increase in recall for both models; while changes in precision are mostly small, recall for positive and negative arguments increases by at least 21pp for both models. As such, *BERT-large$_{topic}$* also beats the *biclstm* by almost 21pp in F_1 score and represents a new state-of-the-art on this dataset.

While the gap to human performance remains at around 18pp in F_1, our proposed approach decreases this gap significantly as compared to the previous state-of-the-art. Based on preliminary experimental results, we suspect that this gap can be further reduced by adding more topics to the training data.

The results show that (1) the *BERT-[base/large]* models largely improve F_1 and precision for arguments and (2) leveraging topic-information yields another strong improvement on the recall of argumentative sentences. The usefulness of topic-information has already been shown by Stab et al.

(2018b) through their *biclstm* and stems from a much higher recall of arguments while losing some of the precision when compared to their *bilstm*. Yet, their approach cannot hold to BERT's superior architecture; the topic-integrating BERT models *BERT-base$_{topic}$* and *BERT-large$_{topic}$* not only compensate for the *biclstm*'s drop in precision, but also increase the recall for *pro* and *con* arguments by at least 18pp and 15pp. We account this performance increase to BERT's multi-head attention between all word pairs, where every word in a sentence has an attention value with the topic (words).

IBM corpus. As a baseline for models that do not use any topic information, we train three simple BiLSTMs with ELMo, BERT, and 300-d GloVe embeddings and compare them to the fine-tuned base and large BERT models. As Table 1 shows, BERT and ELMo embeddings perform around 2.7 and 3.7pp better in accuracy than the GloVe embeddings. *BERT-base* yields even 7pp higher accuracy, while its difference to the large model is only +1pp.

Both *BERT-base* and *BERT-large* outperform the baseline *IBM* set by Shnarch et al. (2018) already by more than 6pp in accuracy[5]. The topic integrating models *IBM$_{ELMo}$* and *IBM$_{BERT}$* do not improve much over their BiLSTM counterparts, which do not use any topic information. Similar to the conclusion for the UKP corpus, we attribute this to the different embedding vectors we retrieve for a topic as compared to the vectors for a topic mention within a sentence. *BERT-base$_{topic}$* and *BERT-large$_{topic}$* show the largest improvement with 8pp over the baseline and represent a new state-of-the-art on this dataset. The fine-tuned BERT models show vast improvements over the baseline, which is on par with the findings for the UKP corpus.

Yet, in contrast to the results on the UKP corpus, adding topic information to the fine-tuned BERT models has only a small effect on the score. This can be explained with the different composition of both corpora: while sentences in the UKP corpus may only be implicitly connected to their related topic (only 20% of all sentences contain their related topic), sentences in IBM's corpus all contain their related topic and are thus explicitly

---

[5] Please note that we refer to our reproduced baseline. Also, the original baseline's performance by Shnarch et al. (2018) can only be guessed, since the numbers are drawn from a figure and do not appear in the text.

Model	UKP Corpus					IBM
	F_1	P_{arg+}	P_{arg-}	R_{arg+}	R_{arg-}	Accuracy
Without topic information						
bilstm (Stab et al., 2018b)	.3796	.3484	.4710	.0963	.2181	.7201
bilstm$_{ELMo}$	.4382	.4639	.5088	.1840	.2778	.7574
bilstm$_{BERT}$	.4631	.5051	.5079	.2074	.3076	.7476
BERT-base	.4680	.5521	.5397	.2352	.2800	.7928
BERT-large	.5019	**.5844**	.5818	.2917	.3154	.8021
With topic information						
outer-att (Stab et al., 2018b)	.3873	.3651	.4696	.1042	.2381	-
biclstm (Stab et al., 2018b)	.4242	.2675	.3887	.2817	.4028	-
biclstm$_{ELMo}$	.3924	.2372	.4381	.0317	.3955	-
biclstm$_{BERT}$	.4243	.3431	.4397	.1060	.4275	-
IBM (Shnarch et al., 2018)	-	-	-	-	-	$\sim$ .74
IBM (*reproduced*)	-	-	-	-	-	.7288
IBM$_{ELMo}$	-	-	-	-	-	.7651
IBM$_{BERT}$	-	-	-	-	-	.7480
BERT-base$_{topic}$	.6128	.5048	.5313	.4698	**.5795**	**.8137**
BERT-large$_{topic}$	**.6325**	.5535	**.5843**	**.5051**	.5594	.8131
Human Performance	.8100	-	-	-	-	-

Table 1: Results of each model for sentence-level argument classification using cross-topic evaluation on the UKP Sentential Argument Mining Corpus and on the IBM Debater® - Evidence Sentences dataset. Blank fields result from dataset-specific models. **P**: precision, **R**: recall, **arg+**: *pro*-arguments, **arg-**: *con*-arguments.

connected to it (although topics are masked with a placeholder). Hence, in the IBM corpus, there is much less need for the additional topic information in order to recognize the relatedness to a sentence.

5 Argument Clustering

Having identified a large amount of argumentative text for a topic, we next aim at grouping the arguments talking about the same aspects.

For any clustering algorithm, a meaningful similarity between argument pairs is crucial and needs to account for the challenges regarding argument aspects, e.g., different aspect granularities, context-dependency or aspect multiplicity. Another requirement is the robustness for topic-dependent differences.

Therefore, in this section, we study how sentence-level argument similarity and clustering can be improved by using contextualized word embeddings. We evaluate our methods on the *UKP ASPECT* and the AFS corpus (see Section 3.2).

5.1 Clustering Method

We use agglomerative hierarchical clustering (Day and Edelsbrunner, 1984) to cluster arguments.

We use the average linkage criterion to compute the similarity between two cluster A and B: $\frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} d(a, b)$, for a given similarity metric d. As it is a priori unknown how many dif-

ferent aspects are discussed for a topic (number of clusters), we apply a stopping threshold which is determined on the train set.

We also tested the k-means and the DBSCAN clustering algorithms, but we found that agglomerative clustering generally yielded better performances in preliminary experiments.

Agglomerative clustering uses a pairwise similarity metric d between arguments. We propose and evaluate various similarity metrics in two setups: (1) Without performing a clustering, i.e. the quality of the metric is directly evaluated (*without clustering* setup), and (2) in combination with the described agglomerative clustering method (*with clustering* setup).

5.2 Experimental Setup

We differentiate between unsupervised and supervised methods. Our unsupervised methods include no pre-training whereas the supervised methods use some data for fine-tuning the model. For the UKP ASPECT corpus, we binarize the four labels to only indicate similar and dissimilar argument pairs. Pairs labeled with *some* and *high similarity* were labeled as *similar*, pairs with *no similarity* and *different topic* as *dissimilar*.

We evaluate methods in a 4-fold cross-validation setup: seven topics are used for testing and 21 topics are used for fine-tuning. Final evaluation results are the average over the four folds. In case of supervised clustering methods, we use 17 topics for training and four topics for tuning. In

	Without Clustering			With Clustering		
Model	$\mathbf{F_{mean}}$	$\mathbf{F_{sim}}$	$\mathbf{F_{dissim}}$	$\mathbf{F_{mean}}$	$\mathbf{F_{sim}}$	$\mathbf{F_{dissim}}$
Human Performance	.7834	.7474	.8194	.7070	.6188	.7951
Random predictions	.4801	.3431	.6171	.4253	.3162	.5344
Unsupervised Methods						
Tf-Idf	.6118	.5230	.7007	.5800	.4892	.6708
InferSent - fastText	.6621	.5866	.7376	.6344	.5443	.7584
InferSent - GloVe	.6494	.5472	.7517	.6149	.4587	.7711
GloVe Embeddings	.6468	.5632	.7304	.5926	.4605	.7246
ELMo Embeddings	.6447	.5355	.7538	.6366	.5347	.7384
BERT Embeddings	.6539	.5232	.7848	.6070	.4818	.7323
Supervised Methods: Cross-Topic Evaluation						
BERT-base	**.7401**	**.6695**	.8107	.7007	**.6269**	.7746
BERT-large	.7244	.6297	**.8191**	**.7135**	.6125	**.8146**

Table 2: F_1 scores on the UKP ASPECT Corpus.

their experiments on the AFS corpus, Misra et al. (2016) only performed a within-topic evaluation by using 10-fold cross-validation. As we are primarily interested in cross-topic performances, we evaluate our methods also cross-topic: we train on two topics, and evaluate on the third.

5.3 Evaluation

For the UKP ASPECT dataset we compute the marco-average F_{mean} for the F_1-scores for the `similar`-label (F_{sim}) and for the `dissimilar`-label (F_{dissim}).

In the *without clustering* setup, we compute the similarity metric ($d(a, b)$) for an argument pair directly, and assign the label `similar` if it exceeds a threshold, otherwise `dissimilar`. The threshold is determined on the train set of a fold for unsupervised methods. For supervised methods, we use a held-out dev set.

In the *with clustering* setup, we use the similarity metric to perform agglomerative clustering. This assigns each argument exactly one cluster ID. Arguments pairs in the same cluster are assigned the label `similar`, and argument pairs in different clusters are assigned the label `dissimilar`. We use these labels to compute F_{sim} and F_{dissim} given our gold label annotations.

For the AFS dataset, Misra et al. (2016) computed the correlation between the predicted similarity and the annotated similarity score. They do not mention which correlation method they used. In our evaluation, we show Pearson correlation (r) and Spearman's rank correlation coefficient (ρ).

5.4 Similarity Metrics

We experiment with the following methods to compute the similarity between two arguments.
Tf-Idf. We computed the most common words (without stop-words) in our training corpus and compute the cosine similarity between the Tf-Idf vectors of a sentence.

InferSent. We compute the cosine-similarity between the sentence embeddings returned by InferSent (Conneau et al., 2017).

Average Word Embeddings. We compute the cosine-similarity between the average word embeddings for GloVe, ELMo and BERT.

BERT. We fine-tune the BERT-uncased model to predict the similarity between two given arguments. We add a sigmoid layer to the special *[CLS]* token and trained it on some of the topics. We fine-tuned for three epochs, with a learning rate of 2e-5 and a batch-size of 32.

Human Performance. We approximated the human upper bound on the UKP ASPECT corpus in the following way: we randomly split the seven pair-wise annotations in two groups, computed their corresponding MACE (Hovy et al., 2013) scores and calculated F_{sim}, F_{dissim} and F_{mean}. We repeated this process ten times and averaged over all runs (*without clustering* setup). For the *with clustering* setup, we applied agglomerative hierarchical clustering on the MACE scores of one of both groups and computed the evaluation metrics using the other group as the gold label. For the AFS dataset, Misra et al. (2016) computed the correlation between the three human annotators.

5.5 Results and Analysis

Unsupervised Methods. Table 2 shows the performance on the novel UKP ASPECT Corpus. When evaluating the argument similarity metrics directly (*without clustering* setup), we notice no large differences between averaging GloVe, ELMo or BERT embeddings. These three setups perform worse than applying InferSent with fast-

	Average	
	r	ρ
Human Performance	.6767	-
Unsupervised Methods		
Tf-Idf	.4677	.4298
InferSent - fastText	.2519	.2423
InferSent - GloVe	.2708	.2663
GloVe Embeddings	.3240	.3400
ELMo Embeddings	.2827	.2675
BERT Embeddings	.3539	.3507
Supervised Methods: Within-Topic Evaluation		
SVR (Misra et al., 2016)	.6333	-
BERT-base	.7475	.7318
BERT-large	.7256	.6959
Supervised Methods: Cross-Topic Evaluation		
BERT-base	.5849	.5723
BERT-large	.6202	.6034

Table 3: Pearson correlation r and Spearman's rank correlation ρ on the AFS dataset (Misra et al., 2016) averaged over the three topics.

Text embeddings. *Tf-Idf* shows the worst performance. In Table 3, we show the performances for the AFS corpus (detailed results in the appendix, Table 5). In contrast to the ASPECT Corpus, the *Tf-Idf* method achieves the best performance and *InferSent - fastText* embeddings achieved the worst performance. As for the ASPECT Corpus, ELMo and BERT embeddings do not lead to an improvement compared to averaged GloVe embeddings.

Unsupervised methods compute some type of similarity between sentence pairs. However, as our experiments shows, this similarity notion is not necessarily the notion needed for the task.

Supervised Methods. We fine-tune the BERT model for some of the topics and study the performance on unseen topics. For the ASPECT Corpus, we observe a performance increase of 7.8pp. Identifying dissimilar arguments (F_{dissim}) is on-par with the human performance, and identifying similar arguments achieves an F-score of .67, compared to .75 for human annotators. For the AFS dataset, we observe that fine-tuning the BERT model significantly improves the performance by 11pp compared to the previous state-of-the-art from Misra et al. (2016).

In a cross-topic evaluation setup on the AFS dataset, we observe that the performance drops to .57 Spearman correlation. This is still significantly larger than the best unsupervised method.

We evaluated the effect of the training set size on the performance of the BERT model for the ASPECT Corpus. A certain number of topics were randomly sampled and the performance was evaluated on distinct topics. This process was repeated

10 times with different random seeds (Reimers and Gurevych, 2018). Table 4 shows the averaged results.

By allowing fine-tuning on five topics we are able to improve the F_{mean}-score to .71 compared to .65 when using BERT without fine-tuning (*without clustering* setup). Adding more topics then slowly increases the performance.

	w/o Clustering	With Clustering
#Topics	F_{mean}	F_{mean}
1	0.6244	0.5943
3	0.6817	0.6322
5	0.7134	0.6563
7	0.7164	0.6703
9	0.7151	0.6697
11	0.7305	0.6988
13	0.7350	0.6964
15	0.7370	0.7010
17	0.7401	0.7034

Table 4: F_1 scores on the UKP ASPECT Corpus with increasing training set sizes (BERT model).

With Clustering. We studied how the performance changes on the ASPECT corpus if we combine the similarity metric with agglomerative clustering (Table 2). We notice that the performances drop by up to 7.64pp. Agglomerative clustering is a strict partitioning algorithm, i.e., each object belongs to exactly one cluster. However, an argument can address more than one aspect of a topic, therefore, arguments could belong to more than one cluster. Hence, strict partitioning clustering methods introduce a new source of errors.

We can estimate this source of error by evaluating the transitivity in our dataset. For a strict partitioning setup, if argument A $\sim$ B, and B $\sim$ C are similar, then A $\sim$ C are similar. This transitivity property is violated in 376 out of 1,714 (21.9%) cases, indicating that strict partitioning is a suboptimal setup for the ASPECT dataset. This also explains why the human performance in the *with clustering* setup is significantly lower than in the *without clustering* setup. As Table 2 shows, a better similarity metric must not necessarily lead to a better clustering performance with agglomerative clustering. Humans are better than the proposed BERT-model at estimating the pairwise similarity of arguments. However, when combined with a clustering method, the performances are on-par.

6 Conclusion

Open-domain argument search, i.e. identifying and aggregating arguments for unseen topics, is

a challenging research problem. The first challenge is to identify suitable arguments. Previous methods achieved low F_1-scores in a cross-topic scenario, e.g., Stab et al. (2018b) achieved an F_1-score of .27 for identifying *pro*-arguments. We could significantly improve this performance to .53 by using contextualized word embeddings. The main performance gain came from integrating topic information into the transformer network of BERT, which added 13pp compared to the setup without topic information.

The second challenge we addressed is to decide whether two arguments on the same topic are similar. Previous datasets on argument similarity used curated lists of arguments, which eliminates noise from the argument classification step. In this publication, we annotated similar argument pairs that came from an argument search engine. As the annotation showed, about 16% of the pairs were noisy and did not address the target topic.

Unsupervised methods on argument similarity showed rather low performance scores, confirming that fine-grained semantic nuances and not the lexical overlap determines the similarity between arguments. We were able to train a supervised similarity function based on the BERT transformer network that, even with little training data, significantly improved over unsupervised methods.

While these results are very encouraging and stress the feasibility of open-domain argument search, our work also points to some weaknesses of the current methods and datasets. A good argument similarity function is only the first step towards argument clustering. We evaluated the agglomerative clustering algorithm in combination with our similarity function and identified it as a new source of errors. Arguments can address multiple aspects and therefore belong to multiple clusters, something that is not possible to model using partitional algorithms. Future work should thus study the overlapping nature of argument clustering. Further, more realistic datasets, that allow end-to-end evaluation, are required.

Acknowledgments

References

Filip Boltužić and Jan Šnajder. 2015. Identifying prominent arguments in online debates using semantic textual similarity. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 110–115.

Elena Cabrio and Serena Villata. 2012. Natural Language Arguments: A Combined Approach. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 205–210.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

William H. E. Day and Herbert Edelsbrunner. 1984. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, 1(1):7–24.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

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

Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. 2014. Context dependent claim detection. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1489–1500.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*.

Amita Misra, Brian Ecker, and Marilyn A. Walker. 2016. Measuring the similarity of sentential arguments in dialogue. In *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*, pages 276–287.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Nils Reimers and Iryna Gurevych. 2018. Why comparing single performance scores does not allow to draw conclusions about machine learning approaches. *arXiv preprint arXiv:1803.09578*.

Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M. Khapra, Ehud Aharoni, and Noam Slonim. 2015. Show me your evidence - an automatic method for context dependent evidence detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 440–450.

Eyal Shnarch, Carlos Alzate, Lena Dankin, Martin Gleize, Yufang Hou, Leshem Choshen, Ranit Aharonov, and Noam Slonim. 2018. Will it Blend? Blending Weak and Strong Labeled Data in a Neural Network for Argumentation Mining. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 599–605.

Christian Stab, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna 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: System Demonstrations*, pages 21–25.

Christian Stab and Iryna Gurevych. 2014. Identifying Argumentative Discourse Structures in Persuasive Essays. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 46–56.

Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. 2018b. Cross-topic argument mining from heterogeneous sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, volume Long Papers, pages 3664–3674.

Amine Trabelsi and Osmar R Zaïane. 2015. Extraction and clustering of arguing expressions in contentious text. *Data & Knowledge Engineering*, 100:226–239.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Henning Wachsmuth, Nona Naderi, Ivan Habernal, Yufang Hou, Graeme Hirst, Iryna Gurevych, and Benno Stein. 2017. Argumentation quality assessment: Theory vs. practice. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 250–255.

Henning Wachsmuth, Shahbaz Syed, and Benno 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: Long Papers)*, pages 241–251.

A Appendices

A.1 UKP ASPECT Corpus: Amazon Mechanical Turk Guidelines and Inter-annotator Agreement

The annotations required for the UKP ASPECT Corpus were acquired via crowdsourcing on the Amazon Mechanical Turk platform. Workers participating in the study had to be located in the US, with more than 100 HITs approved and an overall acceptance rate of 90% or higher. We paid them at the US federal minimum wage of $7.25/hour. Workers also had to qualify for the study by passing a qualification test consisting of twelve test questions with argument pairs. Figure 2 shows the instructions given to workers.

A.2 AFS Corpus: Detailed Results

Table 5 shows the full results of the (un)supervised methods for the argument similarity calculation on the AFS dataset (all topics).

	Gun Control		Gay Marriage		Death Penalty		Avg.	
	r	ρ	r	ρ	r	ρ	r	ρ
Human Performance	.6900	-	.6000	-	.7400	-	.6767	-
Unsupervised Methods								
Tf-Idf	.6266	.5528	.4107	.3778	.3657	.3589	.4677	.4298
InferSent - fastText	.3376	.3283	.1012	.1055	.3168	.2931	.2519	.2423
InferSent - GloVe	.3757	.3707	.1413	.1435	.2953	.2847	.2708	.2663
GloVe Embeddings	.4344	.4485	.2519	.2741	.2857	.2973	.3240	.3400
ELMo Embeddings	.3747	.3654	.1753	.1709	.2982	.2663	.2827	.2675
BERT Embeddings	.4575	.4460	.1960	.1999	.4082	.4072	.3539	.3507
Supervised Methods: Within-Topic Evaluation								
SVR (Misra et al., 2016)	.7300	-	.5400	-	.6300	-	.6333	-
BERT-base	.8323	.8076	.6255	.6122	.7847	.7768	.7475	.7318
BERT-large	.7982	.7592	.6240	.6137	.7545	.7149	.7256	.6959
Supervised Methods: Cross-Topic Evaluation								
BERT-base	.6892	.6689	.4307	.4236	.6339	.6245	.5849	.5723
BERT-large	.6895	.6749	.5071	.4866	.6641	.6486	.6202	.6034

Table 5: Pearson correlation r and Spearman's rank correlation ρ on the AFS dataset. Within-Topic Evaluation: 10-fold cross-validation. Cross-Topic Evaluation: System trained on two topics, evaluated on the third.

Read each of the following sentence pairs and indicate whether they argue about the same aspect with respect to the given topic (given as "Topic Name" on top of the HIT). There are **four options**, of which one needs to be assigned to each pair of sentences (arguments). Please read the following for more details.

- **Different Topic/Can't decide**: Either one or both of the sentences belong to a topic different than the given one, or you can't understand one or both sentences. If you choose this option, you need to very briefly explain, why you chose it (e.g. "The second sentence is not grammatical", "The first sentence is from a different topic" etc.). For example,

  Argument A: "*I do believe in the death penalty, tit for tat*".

  Argument B: "*Marriage is already a civil right everyone has, so like anyone you have it too*".

- **No Similarity**: The two arguments belong to the same topic, but they don't show any similarity, i.e. they speak about completely different aspects of the topic. For example,

  Argument A: "*If murder is wrong then so is the death penalty*".

  Argument B: "*The death penalty is an inappropriate way to work against criminal activity*".

- **Some Similarity**: The two arguments belong to the same topic, showing semantic similarity on a few aspects, but the central message is rather different, or one argument is way less specific than the other. For example,

  Argument A: "*The death penalty should be applied only in very extreme cases, such as when someone commands genocide*".

  Argument B: "*An eye for an eye: He who kills someone else should face capital punishment by the law*".

- **High Similarity**: The two arguments belong to the same topic, and they speak about the same aspect, e.g. using different words. For example, Argument A: "*An ideal judiciary system would not sentence innocent people*".

  Argument B: "*The notion that guiltless people may be sentenced is indeed a judicial system problem*".

Your rating should not be affected by whether the sentences attack (e.g. "*Animal testing is cruel and inhumane*" for the topic "*Animal testing*") or support (e.g. "*Animals do not have rights, therefore animal testing is fair*" for the topic "*Animal testing*") the topic, but only by the aspect they are using to support or attack the topic.

Figure 2: Amazon Mechanical Turk HIT Guidelines used in the annotation study for the Argument Aspect Similarity Corpus.