Segmenting and Clustering Noisy Arguments

Lorik Dumani(✉), Christin Katharina Kreutz(✉), Manuel Biertz(✉), Alex Witry(✉), and Ralf Schenkel(✉)

Trier University, 54286 Trier, Germany
{dumani,kreutzch,biertz,s4alwitr,schenkel}@uni-trier.de

Abstract. Automated argument retrieval for queries is desirable, e.g., as it helps in decision making or convincing others of certain actions. An argument consists of a claim supported or attacked by at least one premise. The claim describes a controversial viewpoint that should not be accepted without evidence given by premises. Premises are composed of Elementary Discourse Units (EDUs) which are their smallest contextual components. Oftentimes argument search engines find similar claims to a query first before returning their premises. Due to heterogeneous data sources, premises often appear repeatedly in different syntactic forms. From an information retrieval perspective, it is essential to rank premises relevant for a query claim highly in a duplicate-free manner. The main challenge in clustering them is to avoid redundancies as premises frequently address various aspects, i.e., consist of multiple EDUs. So, two tasks can be defined: segmentation of premises in EDUs and clustering of similar EDUs.
In this paper we make two contributions: Our first contribution is the introduction of a noisy dataset with 480 premises for 30 queries crawled from debate portals which serves as a gold standard for the segmentation of premises into EDUs and the clustering of EDUs. Our second contribution consists of first baselines for the two mentioned tasks, for which we evaluated various methods. Our results show that an uncurated dataset is a major challenge and that clustering EDUs is only reasonable with premises as context information.

1 Introduction

Computational argumentation is an important building block in decision making applications. Retrieving supporting and opposing premises for controversial claims can help to make informed decisions on the topic or, when seen from a different viewpoint, to persuade others to take particular standpoints or even actions. In line with existing work in this field, we consider arguments that consist of a claim that is supported or attacked by at least one premise [24]. The claim is the central component of an argument, and it is usually controversial [23]. The premises increase or decrease the claim's acceptance [11]. The stance of a premise indicates if it supports (pro) or attacks (con) the claim. Table 1 shows an example for an argument consisting of a claim supported or opposed by premises.

Table 1. Example of a claim c and its premises p_1, p_2 and p_3.

var.	type	stance	content
c	claim	-	*Aviation fuel should be taxed*
p_1	premise	pro	*Less CO_2 emissions lead to a clean environment*
p_2	premise	con	*Higher taxes would not change anything*
p_3	premise	pro	*It does not matter that the costs for aviation are already high as the environment can be protected by less CO_2 emissions*

In the NLP community researchers either address argument mining, i.e., the analysis of the structure of arguments in natural language texts (see the work of Cabrio and Villata [4] for an overview of recent contributions), or an information-seeking perspective, i.e., the identification of relevant premises associated with a predefined claim [19]. Due to the rapidly increasing need for argumentative queries, established search engines that only retrieve relevant documents will no longer be sufficient. Instead, argument search engines are required that can provide the best pro and con premises for a query claim. In fact, various argument search engines [27,22] have recently been developed. These systems usually work on claims and premises that were either mined from texts beforehand or extracted from dedicated argument websites such as idebate.org. Their workflow usually starts with finding *result claims* similar to the query claim. Then they locate the *result premises* belonging to these claims to present them as output.

However, these systems face a number of challenges since claims and premises are formulated in natural language. First, premises that are semantically (mostly) equivalent occur repeatedly in different textual representations since they appear in different sources, but should be retrieved only once to avoid duplicates. This requires the clustering of similar premises for result presentation. Second, discussions on debate portals, but also in natural language arguments are often not well-structured, such that a single supporting or attacking piece of text can address several aspects and thus should be represented as multiple premises. For example, a sentence supporting the viewpoint that aviation fuel should be taxed could address two aspects, the potential danger for the environment and the current low tax rate on aviation fuel. Directly using such sentences as formal premises, as seen in premise p_3 in Table 1, would make it impossible to retrieve a duplicate-free and complete list of premises.

This issue can be avoided by dividing the premises into their core aspects and clustering them instead of whole premises. In the literature, the smallest contextual components of a text are called *Elementary Discourse Units* (*EDUs*) [24]. Obtaining high quality EDUs [24] from text (discourse segmentation) is a crucial task preceding all efforts in parsing or representing discourses [21]. Thereby, it takes a pragmatic perspective, i.e., links between discourse segments are established not on semantic grounds but on the author's (assumed) intention [17]. For the explorative purposes outlined here, only the concept of EDUs as smallest, non-overlapping units of intra-text-discourse – mostly clauses – is picked up [15].

In this paper we address the aforementioned limitations and deal with the segmentation of textual premises into EDUs and the clustering of EDUs based on their semantic similarity. Contrasting previous research on both of these tasks that worked with manually curated and thus high-quality argument col-

$\text{EDU}_1(p_1)$ = "*Less CO_2 emissions lead to a clean environment*"
$\text{EDU}_1(p_2)$ = "*Higher taxes would not change anything*"
$\text{EDU}_1(p_3)$ = "*It does not matter that the costs for aviation are already high*"
$\text{EDU}_2(p_3)$ = "*as the environment can be protected by less CO_2 emissions*"

Fig. 1. EDUs extracted from premises in Table 1.

lections, we use a dataset that was crawled from debate portals [10]. Unlike other datasets, the premises in this dataset contain a considerably higher number of sentences and often cover multiple aspects (which is at odds with our generally micro-structural approach to arguments). In addition, as an uncurated real-world dataset, it contains many ill-formulated sentences and other defects. Our contribution is two-fold: First we provide a real-life dataset consisting of 480 premises retrieved for 30 query claims that are segmented into 4,752 EDUs. Then, for each query claim the belonging EDUs have been manually clustered by semantic equivalence. Second, we report our first results for the two tasks of EDU identification and EDU clustering on this dataset.

Our proposed method works as follows: for a given set of textual premises returned by an argument search engine for a query claim, we first identify the EDUs for each result. In the second step, we focus on the clustering of EDUs. To accomplish this, we first generate embeddings and then we cluster those with an agglomerative clustering algorithm. As an example, consider Table 1 again. Here, premise p_3 is composed of two EDUs $\text{EDU}_1(p_3)$ and $\text{EDU}_2(p_3)$ (see Figure 1). In addition to that, $\text{EDU}_2(p_3)$ and $\text{EDU}_1(p_1)$ (where $\text{EDU}_1(p_1)$ is the only EDU of p_1) have the same meaning and therefore should be assigned to the same cluster.

The remainder of this paper is structured as follows: Section 2 provides an overview of related work addressing the segmentation of argumentative texts into EDUs and clustering algorithms. In Section 3 the dataset and its manual annotation is described in more detail. Then, in Section 4 we present and evaluate our methods for extraction and clustering of EDUs. Section 5 concludes our work and provides future research directions.

2 Related Work

There is a plethora of research on *discourse segmentation* of text but to the best of our knowledge, existing approaches are designed for curated datasets. A rule-based approach including a post processing step for identification of starts and ends of EDUs was proposed by Carreras et al. [6]. Among other features they utilize chunks tags and sentence patterns. Soricut and Marcu [21] introduced a probabilistic approach based on syntactic parse trees. Tofiloski et al. [25] perform EDU segmentation based on syntactic and lexical features with the goal of capturing only interesting, not all EDUs. Here, every EDU is conditioned to contain a verb. Others suggest a classifier able to decide whether a word is the beginning, middle or end of a nested EDU using features derived from *Part of Speech* (POS) tags, chunk tags or dependencies to the root element [1]. In a recent paper, Trautmann et al. [26] also argue that "*spans of tokens*" rather than whole sentences should be annotated and define this task as Argument Unit

Recognition and Classification. We omit preprocessing of text and utilization of preconditions which is applicable to a supervised scenario as it might flaw an approach based on uncurated data as no guarantees can be made for a real-world, possibly defective, crawled dataset from debate portals.

The *clustering of similar arguments* is still a recent field of research. Boltuzic and Snajder [3] applied Word2Vec [16] with hierarchical clustering for debate portals. Reimers et al. [19] experiment with contextualized word embedding methods such as ELMo [18] and BERT [8] and show that these can be used to classify and cluster topic-dependent arguments. They use hierarchical clustering with a stopping threshold which is determined on the training set to obtain clusters of premises. However, they do not specify a concrete value. Further, Reimers et al. note that premises sometimes cover different aspects. Hence, we divide premises into their EDUs and cluster these instead. Like them, we also use uncurated data and make use of ELMo and BERT. We additionally utilize the embedding methods InferSent [7], and Flair [2]. Contrasting Reimers et al., we only consider relevant premises for the clustering as we intend to start with a step-by-step approach.

3 Dataset and Labeling

We make use of the argumentation dataset introduced in our prior work [10] where we crawled four debate portals and extracted claims with their associated textual premises. In a follow-up work [9], we built a benchmark collection for argument retrieval based on that dataset. In this former work, we picked 232 randomly chosen claims on the topic energy and used them as query claims to pool the most similar result claims retrieved by standard IR methods. In the latter [9], for 30 of these query claims, we collected the premises of all pooled result claims and manually assessed their relevance with respect to the query claim, using a three-fold scale ("very relevant", "relevant", "not relevant"). This resulted in 1,195 tuples of the form (query claim, result claim, result premise, assessment). Following the practice at TREC (Text REtrieval Conference), a premise is relevant if it has at least one relevant EDU, and very relevant if it contains no aspect not relevant to the initial query claim.

In this paper, we only included result premises that were assessed with "very relevant" or "relevant" to keep the effort for manual assessment reasonable. This means we consider 480 tuples for our new dataset. For each of these 480 result premises, the EDUs were identified by one annotator who is a research assistant from political science and has a deep understanding of argumentation theory. For this segmentation, the annotator followed the manual by Carlson and Marcu [5]. This resulted in a total of 4,752 EDUs for the 480 premises (on average 9.9 EDUs per premise), indicating that premises in debate portals usually cover plenty of aspects and segmentation is indispensable for argument retrieval and clustering.

In a next step, the EDUs were manually clustered by identifying semantically equivalent EDUs and putting them in the same cluster. This was done with support of a modified variant of the OVA tool [12] (http://ova.uni-trier.de/) for modeling complex argumentations, which was enhanced to be capable to store text positions. Since EDUs cannot be further divided by definition, clusters were

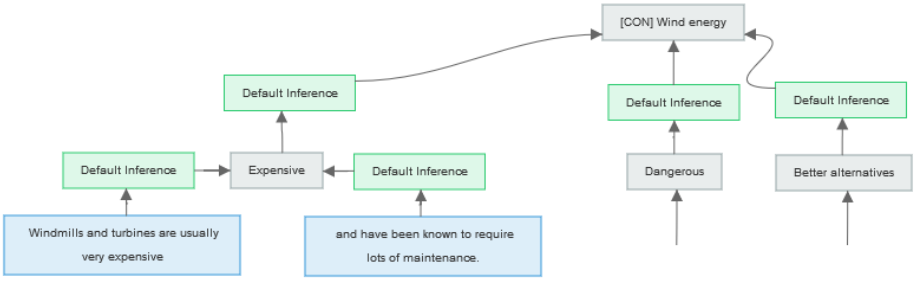

Fig. 2. Screenshot showing an excerpt of the OVA tool used for clustering similar EDUs. The blue nodes represent the EDUs, the gray nodes were added artificially by the annotator and represent the clusters. The green nodes are edges and represent the relations from the EDUs to the clusters they were assigned by the assessor.

formed manually that include all EDUs with the same meaning. For each of the 30 query claims, an OVA view was created where all EDUs identified in result premises for this query were represented as nodes. A human annotator then clustered these nodes by creating an artificial node for each cluster identified and then connecting all semantically identical EDUs to the cluster node by dragging edges. Additionally, to make the clustering more readable, the annotator created three artificial clusters "PRO", "CON", and "CLAIMS" and referenced the previously formed artificial clusters to them depending on their stance with respect to the query. In this paper we will not consider stances. However, since we are making the dataset available (on request), they can be important for further work, for example, for those who also want to use additional distinctions according to the stance.

Figure 2 illustrates a screenshot of the clustering annotation tool. Not all EDUs could plausibly be treated as a single premise (e.g., EDUs that are post-modifiers to noun phrases), thus we also allowed to mark EDUs as context information for other EDUs. For the clustering task, we clustered 1,044 EDUs for 11 queries, distributed to 622 clusters. Because of time constraints, we did not manage to cluster all EDUs of all 30 queries here, and instead only analyzed 11, which are after all more than 1,000 clustered EDUs. The annotators' feedback was that the visualization helped to keep an overview as there were almost 100 EDUs per query to cluster.

4 Methodology and Evaluation

This section describes our approaches for segmenting premises into EDUs and clustering them, as well as an evaluation of the performance of these methods with respect to the ground truth. Figure 3 provides a schematic overview of the different steps. In general, our approach will retrieve clusters of EDUs for input query claims. Given a query claim q_i as well as similar result claims $c_{i,j}$ with associated premises $p_{i,j,k}$. These relevant result claims are retrieved by

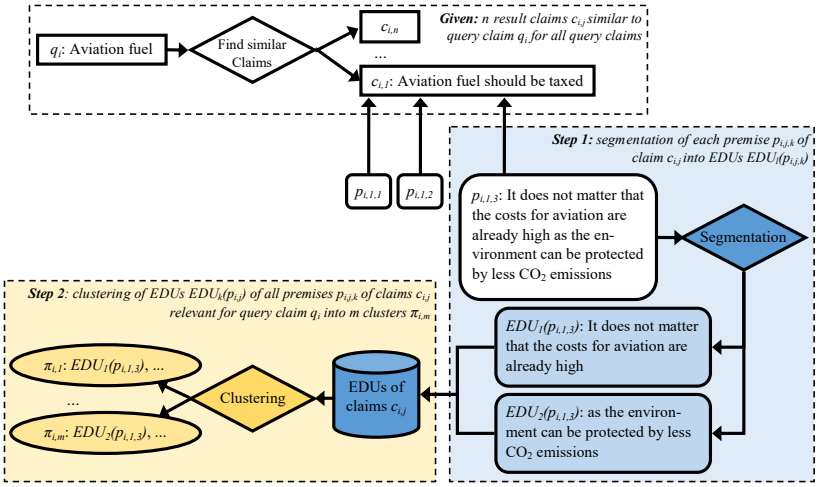

Fig. 3. Schematic overview of the two steps segmentation and clustering.

application of our prior work [10]. In the first step of this approach, premises are divided into EDUs, in the second step all EDUs of premises linked to result claims for our query claim will be clustered.

4.1 Step 1: Segmentation of Premises into EDUs

We first compare different approaches for segmentation of premises into EDUs to the ground truth segmentation from the 30 claims. We focus on basic segmentation methods generating sequential, i.e., non-overlapping EDUs in order to obtain insight into their performance on a real-world dataset as they are often used as a preprocessing step in more sophisticated segmenters [6,21,25,1].

As an initial baseline (*sentence baseline*), we split premises into sentences with CoreNLP (stanfordnlp.github.io/CoreNLP) and considered each sentence as an EDU. As CoreNLP also allows to extract a text's PennTree, which contains the POS tag for each term and displays the closeness of the terms in a hierarchical structure, we also identified EDUs by cutting the PennTree of premises (*tree cut*) at height cutoffs from 1 to 10, denoted by $tc_{i,1\leq i\leq 10}$ in the following. Additionally, we obtained subclauses from sentences which we also regarded as EDUs by applying Tregex (nlp.stanford.edu/software/tregex) (*subclauses*). We also implemented a rule-based splitter (*splitter*) which does consider the peculiarities of our dataset but differs from the ground truth [5]. This splitter is kind of an extension of the sentence baseline, thus sentence boundaries and all kinds of punctuation marks are seen as discourse boundaries [21] so those are used to split premises into EDUs. Further, before conjunctions and terms or phrases indicating subclauses, boundaries are included.

Table 2 shows the performance of the different approaches which we compare in terms of their precision, recall, F_1 score, specificity, and accuracy. Out of the tree cut approaches, tc_3, i.e., the tree cut with cutoff at height 3, obtained the

Table 2. Performance of the different methods to split text to EDUs. Precision, recall, F_1 score, specificity and accuracy.

Method	Prec	Rec	F_1	Spec	Acc
SENTENCE BASELINE	0.3289	**0.9561**	0.4739	0.9140	0.4883
TREE CUT tc_3	0.6223	0.7064	0.6458	0.9472	0.5073
SUBCLAUSES	0.4150	0.3796	0.3905	0.9167	0.4934
SPLITTER	**0.8523**	0.562	**0.6654**	**0.9768**	**0.5244**

highest F_1 score. The rule-based splitter achieved the highest F_1 score of all tested methods. This method results in a high precision combined with a lower recall which is a property of conservative approaches [25]. The comparably poor results for the other approaches may occur since classical preprocessing steps are unfit for approximating human annotations on uncurated real-world datasets. A Kruskal-Wallis test[1] on F_1 scores for boundaries of EDUs computed for each premise of the sentence baseline, splitter and tc_3 holds for $p = 0.05$. Thus, the splitter method is significantly better than the other methods.

Evaluation of EDUs In order to evaluate the quality of EDUs obtained by the annotators as well as our best approaches we constructed triples of the form ($\text{EDU}_{ground\_truth}$, EDU_{tc_3}, $\text{EDU}_{splitter}$) for 50 randomly chosen premises. Within each triple, the EDUs were ranked by their subjective perceived quality by a reviewer who is an expert in computational argumentation and familiar with argumentation theory. Note that it was not shown to the assessor how each EDU was determined and the ordering within triples was shuffled. The expert assessor assigned ranks from 1 to 3 with 1 being the best, ties were permitted.

The ground truth achieved an average rank of 1.66 (#1: 22 times, #2: 23 times, #3: 5 times), tc_3 did perform equally well (#1: 23 times, #2: 21 times, #3: 6 times). The splitter method performed considerably worse with an average rank of 2.64 (#1: 6 times, #2: 6 times, #3: 38 times). As the ground truth would be expected to outperform other approaches clearly, this outcome indicates firstly the difficulty in the annotation process, secondly the subjective perception of what is better and what is less good, and thirdly the difficulty in correctly capturing language with computers. Figure 4 shows an example of both the manually created EDUs and those created by the splitter method.

4.2 Step 2: Clustering of EDUs

In order to build clusters of EDUs automatically for each of the eleven claims, first we obtained the embedding vectors of EDUs using ELMo, BERT, Flair, and InferSent[2]. For this task, we consider the segmentation of premises into

[1] A Kruskal-Wallis test was used as data in the three groups is not normally distributed; this was tested with a Shapiro-Wilk test.

[2] We used the implementations provided by https://github.com/facebookresearch/InferSent and https://github.com/flairNLP/flair.

EDUs by ground truth:
[From what I understand,] [the cheap oil is something] [that will not only effect the economy in the long run,] [but it will also hurt those] [who want to receive retirement or disability benefits at the federal level.] [It's great to finally have cheaper gas than that] [which was nearly $3 in the past.] [I do think] [it might have an adverse effect on our economy.]

EDUs by splitter:
[From what I understand,] [the cheap oil is something] [that will not only effect the economy in the long run,] [but it will also hurt those] [who want] [to receive retirement] [or disability benefits at the federal level.] [It's great] [to finally have cheaper gas] [than] [that] [which was nearly $3 in the past.] [I do think it might have an adverse effect on our economy.]

Fig. 4. Example of EDUs manually created and those by the method splitter. EDUs are encompassed by square brackets. Differently identified EDUs between ground truth and method splitter are underlined.

EDUs given by the ground truth of Section 4.1. Otherwise, an automatic external evaluation would be infeasible. We derived eight vectors per EDU and embedding technique by extending EDUs with context information, i.e., we obtained tuples (EDU, ctx) with context ctx from all combinations of the power set $\mathcal{P}(\{premise, result\ claim, query\ claim\})$. After that, we performed an agglomerative (hierarchical) clustering of the EDUs of all claims related to the query for each of the eleven queries as it is the state-of-the-art for clustering arguments [3,19]. Then, since we do not know the number of clusters a priori, we performed a dynamic tree cut [14]. The advantage of this approach over other approaches such as k-means is that there is no need to specify a final number of result clusters, which is not known in our case. The benefit of agglomerative clustering over divisible clustering is certainly the lower runtime. As a straightforward baseline, all EDUs from the same premise are assigned to the same cluster ($BL_{premiseAsCluster}$). Two additional baselines consist of one big cluster containing all EDUs ($BL_{oneCluster}$), as well as many clusters, each containing one EDU ($BL_{ownClusters}$). The quality of the clustering was measured with external and internal evaluation measures. While external evaluation measures base on previous knowledge, in our case the ground truth clustering formed by the assessor, the internal evaluation measures base on information that only involves the vectors of the datasets themselves [20].

With regard to the external cluster evaluation metrics, we measured the following three: the *purity*, the *adjusted mutual information* (AMI), and the *adjusted Rand index* (Rand). For the internal cluster evaluation, we measured the *Calinski-Harabasz index* (CHI) and the *Davies-Bouldin index* (DBI).[3] Concise descriptions of the metrics can be found in Table 3. The results of the external and internal evaluations can be found in Table 4.

We can observe that $BL_{premiseAsCluster}$ outperforms all methods for the external evaluation measures except for the perfect purity of $BL_{ownClusters}$. In general $BL_{oneCluster}$ and $BL_{ownClusters}$ do not produce surprising results for the external evaluation. CHI and DBI are undefined for their number of clusters.

[3] We used the implementations provided by https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics.cluster.

Table 3. Descriptions of internal and external cluster evaluation metrics.

Type	Name	Brief description
external	purity	Measures the extent to which clusters contain a single class. Its value ranges from 0 to 1, with 1 being the best. Generally, a high (compared to the number of clustered entities) number of clusters results in a high purity.
external	AMI	Measures the mutual dependence between two random variables and qualifies the amount of information obtained about one random variable through observing the other random variable. Values are adjusted for chance. Its values range between 0 and 1, where 1 implies a perfect correlation.
external	Rand	Computes the accuracy and ranges from 0 to 1. It penalizes both false positive and false negative decisions with equal weights during clustering. Values are adjusted for chance.
internal	CHI	Rate between inter-cluster and intra-cluster dispersion. Higher values suggest dense and well-separated clusters. The number of clusters must lie between 2 and |data points| -1.
internal	DBI	Shows the average similarity of each cluster to the most similar cluster. Clusters further apart from each other produce better results. The lowest possible and best score is 0. The number of clusters must lie between 2 and |data points| -1.

It is remarkable that the methods that perform best have the corresponding premises as additional context information, while the worst performing methods do not utilize them. In fact, for each evaluation measure, all 16 methods that include the premise as context information achieve better values than those 16 that do not include it. The inclusion of a query claim and result claim as context information seems to have no influence on the ranking, because the methods with and without usage of this context information in the ranking are sometimes better and sometimes worse. Thus, clustering EDUs always requires the context information in the premise. Kruskal-Wallis tests with $p = 0.05$ were conducted on the three external and two internal measures of the eleven query claims of the three best performing methods as well as the baseline.[4] For AMI, CHI and DBI significant differences were found. For purity and Rand, no significant differences could be found between the four groups.

For the internal cluster evaluations all methods that include the premise as context information produce better outcomes than those computed for the baseline $BL_{premiseAsCluster}$ clusters. The best values were achieved when using EDUs computed with ELMo or BERT embeddings. This observation clearly shows challenges in automatic clustering of arguments in difficult datasets. We conducted Mann-Whitney U tests on the five measures from the eleven clusterings for each of the three best methods and their counterpart without utilization of the premise as context-information (e.g. $\text{ELMo}_{e,p,q}$ and $\text{ELMo}_{e,q}$ were observed as a pair) with $p = 0.05$.[5] We found significant differences in values for purity, AMI, Rand, CHI and DBI for ELMo as well as InferSent; for the two experiments with BERT embeddings, significant differences were found for all

[4] Kruskal-Wallis tests were used as except for purity, data is not normally distributed in the four groups; this was tested with Shapiro-Wilk tests.

[5] Mann-Whitney U tests were used as for all pairs, some of the measures are not normally distributed; this was tested with Shapiro-Wilk tests.

Table 4. The **external** and **internal** clustering evaluation including: mean *purity*, mean *adjusted mutual information (AMI)*, mean *adjusted Rand index* (*Rand*), mean *Calinski-Harabasz index* (*CHI*), and mean *Davies-Bouldin index* (*DBI*) for the baselines $BL_{premiseAsCluster}$, $BL_{oneCluster}$, $BL_{ownClusters}$ (see Section 4.2), for the best (marked bold) as well as worst (underlined) performing combinations of context (premise p, result claim r, query claim q) with EDU e and embedding methods for the 11 queries.

	External			Internal	
Method	**purity**	**AMI**	**Rand**	**CHI**	**DBI**
$BL_{premiseAsCluster}$	0.6281	**0.4863**	**0.3618**	1.337	2.882
$BL_{oneCluster}$	0.2512	0	0	-	-
$BL_{ownClusters}$	**1**	0	0	-	-
$\text{ELMo}_{e,p,q}$	**0.6032**	0.3453	0.2406	**6.4446**	2.4161
$\text{InferSent}_{e,p}$	0.5977	**0.3888**	**0.2837**	4.3765	2.5958
$\text{BERT}_{e,p,r}$	0.5996	0.3492	0.2496	4.2647	**2.3412**
$\text{InferSent}_{e,q}$	0.4309	0.046	0.0255	1.2496	3.1276
$\text{Flair}_{e,q}$	0.4315	0.0465	0.023	1.3228	3.1455
Flair_{e}	0.4492	0.0695	0.0477	1.2346	3.158

external measures and CHI. From this observation we derive the usefulness of premises as context information for the overall clustering quality.

Error Analysis of the Clustering We performed an additional manual evaluation of the clustering by including the three best performing methods shown in Table 4, as well as the initial manual clustering. For this evaluation we randomly picked 30 clusters which contain at least three EDUs per cluster (120 clusters in total) and added a new EDU to each of them, which two human annotators (different from the one who constructed the ground truth in Section 3) had to spot to determine the perceived soundness of the clustering. This new EDU originated from the same premise or, if no EDU was available there, from the same query. For each cluster at most five EDUs were shown. They were shuffled and the new EDU was placed at a random position. Additionally, we include a random baseline. Here, for each of the 120 evaluation clusters, the intruding EDU was picked at random.

Only the query and the EDUs were presented to the annotators. For the manually labeled clusters, both annotators managed to identify 16 out of 30 false EDUs. For $\text{InferSent}_{e,p}$, $\text{BERT}_{e,p,r}$, and $\text{ELMo}_{e,p,q}$, it was 11.5, 8.5, and 7 out of 30 on average, respectively.[6] The inter-annotator agreement, calculated with Krippendorff's α [13], was 0.463 on a nominal scale, implying that the agreement is moderate. The random baseline picked a total of 9, 4, and 6 wrong EDUs. We found no significant differences with Kruskal-Wallis tests for clustering based on BERT, Elmo and InferSent embeddings for the number of correctly identified intruding EDUs by the two annotators and the random baseline. Yet, for the ground truth, significant differences were found. The results show that

[6] The differences in the annotations were two times 1, once 0, and once 4.

the automatic clustering of EDUs by semantics still lags behind manual annotation. However, they also reveal that even the manually produced clustering is ambiguous, as one would have expected to find (almost) all the wrong EDUs. Overall, the annotators' impression was that it was a very difficult task to spot the intruding EDU because except for the query no context information was given. In most cases, the query did not really help in identifying the out-of-place EDU. In contrast, when creating the ground truth, the (other) annotator first read the whole texts associated with result claims and then decided which EDUs should be clustered. This is an important difference.

5 Conclusion and Future Work

Segmenting complex premises and clustering of semantically similar premises are important tasks in the retrieval of arguments, as argument retrieval systems need to deal with complex natural-language statements and should not show duplicate results. This is even a problem for arguments extracted from debate portals since single textual premises often address a variety of aspects. In this paper we discussed the segmentation of premises into EDUs, as well as clustering these from an uncurated dataset. Our results show that segmenting premises into their EDUs in such a dataset with rule-based procedures that are suitable for curated datasets is feasible, in particular by following either a precision or a recall-oriented approach. Furthermore, we have seen that clustering EDUs only performs comparably well with the associated premises as context information at least. The segmentation of EDUs from noisy texts remains a difficult task for now. We provide the labeled data of EDUs and clusters of EDUs so that future argument mining methods can use it for evaluation of their performance.

Future work will include extracting unique EDUs using context information and further analyzing properties of real-world datasets which impede manual EDU extraction and clustering. With these insights, an annotation support system could be constructed to help manually identifying and clustering EDUs.

Acknowledgments We would like to thank Anna-Katharina Ludwig for her invaluable help in clustering the EDUs and Patrick J. Neumann for his help in the implementation.

This work has been funded by the Deutsche Forschungsgemeinschaft (DFG) within the project ReCAP, Grant Number 375342983 - 2018-2020, as part of the Priority Program "Robust Argumentation Machines (RATIO)" (SPP-1999).

References

1. Afantenos, S.D., Denis, P., Muller, P., Danlos, L.: Learning recursive segments for discourse parsing. In: LREC (2010)
2. Akbik, A., Blythe, D., Vollgraf, R.: Contextual string embeddings for sequence labeling. In: COLING (2018)
3. Boltuzic, F., Snajder, J.: Identifying prominent arguments in online debates using semantic textual similarity. In: ArgMining@HLT-NAACL (2015)

4. Cabrio, E., Villata, S.: Five years of argument mining: a data-driven analysis. In: IJCAI (2018)
5. Carlson, L., Marcu, D.: Discourse Tagging Reference Manual, https://www.isi.edu/~marcu/discourse/tagging-ref-manual.pdf
6. Carreras, X., Màrquez, L.: Boosting trees for clause splitting. In: ACL (2001)
7. Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A.: Supervised learning of universal sentence representations from natural language inference data. In: EMNLP (2017)
8. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)
9. Dumani, L., Neumann, P.J., Schenkel, R.: A framework for argument retrieval - ranking argument clusters by frequency and specificity. In: ECIR. LNCS, vol. 12035. Springer (2020)
10. Dumani, L., Schenkel, R.: A systematic comparison of methods for finding good premises for claims. In: SIGIR (2019)
11. van Eemeren, F.H., Garssen, B., Krabbe, E.C.W., Henkemans, A.F.S., Verheij, B., Wagemans, J.H.M. (eds.): Handbook of Argumentation Theory. Springer (2014)
12. Janier, M., Lawrence, J., Reed, C.: OVA+: an argument analysis interface. In: COMMA (2014)
13. Krippendorff, K.: Estimating the reliability, systematic error and random error of interval data. Educational and Psychological Measurement **30** (1970)
14. Langfelder, P., Zhang, B., Horvath, S.: Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R. Bioinformatics **24**(5) (11 2007)
15. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: Toward a functional theory of text organization. Text & Talk **8**(3) (1988)
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NeurIPS (2013)
17. Peldszus, A., Stede, M.: From Argument Diagrams to Argumentation Mining in Texts. IJCINI **7**(1) (2013)
18. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: NAACL-HLT (2018)
19. Reimers, N., Schiller, B., Beck, T., Daxenberger, J., Stab, C., Gurevych, I.: Classification and clustering of arguments with contextualized word embeddings. In: ACL (2019)
20. Rendón, E., Abundez, I., Arizmendi, A., Quiroz, E.M.: Internal versus external cluster validation indexes. Int. J. Comput. Commun. **5**(1) (2011)
21. Soricut, R., Marcu, D.: Sentence level discourse parsing using syntactic and lexical information. In: HLT-NAACL (2003)
22. Stab, C., Daxenberger, J., Stahlhut, C., Miller, T., Schiller, B., Tauchmann, C., Eger, S., Gurevych, I.: Argumentext: Searching for arguments in heterogeneous sources. In: NAACL-HTL (2018)
23. Stab, C., Gurevych, I.: Identifying argumentative discourse structures in persuasive essays. In: EMNLP (2014)
24. Stede, M., Afantenos, S.D., Peldszus, A., Asher, N., Perret, J.: Parallel discourse annotations on a corpus of short texts. In: LREC (2016)
25. Tofiloski, M., Brooke, J., Taboada, M.: A syntactic and lexical-based discourse segmenter. In: ACL and AFNLP (2009)
26. Trautmann, D., Daxenberger, J., Stab, C., Schütze, H., Gurevych, I.: Fine-grained argument unit recognition and classification. In: AAAI (2020)
27. Wachsmuth, H., Potthast, M., Khatib, K.A., Ajjour, Y., Puschmann, J., Qu, J., Dorsch, J., Morari, V., Bevendorff, J., Stein, B.: Building an argument search engine for the web. In: ArgMining@EMNLP (2017)