Clustering of Argument Graphs Using Semantic Similarity Measures

Karsten Block, Simon Trumm, Premtim Sahitaj, Stefan Ollinger, and Ralph Bergmann(✉)

Business Information Systems II, University of Trier, 54296 Trier, Germany
{s4kabloc,s4sitru,sahitaj,ollinger,bergmann}@uni-trier.de
http://www.wi2.uni-trier.de

Abstract. Research on argumentation in Artificial Intelligence recently investigates new methods that contribute to the vision of developing robust argumentation machines. One line of research explores ways of reasoning with natural language arguments coming from information sources on the web as a foundation for the deliberation and synthesis of arguments in specific domains. This paper builds upon arguments represented as argument graphs in the standardized Argument Interchange Format. While previous work was focused on the development of semantic similarity measures used for the case-based retrieval of argument graphs, this paper addresses the problem of clustering argument graphs to explore structures that facilitate argumentation interpretation. We propose a k-medoid and an agglomerative clustering approach based on semantic similarity measures. We compare the clustering results based on a graph-based semantic measure that takes the structure of the argument into account with a semantic word2vec measure on the pure textual argument representation. Experiments based on the Microtext corpus show that the graph-based similarity is best on internal evaluation measures, while the pure textual measure performs very well for identifying topic-specific clusters.

Keywords: Argumentation · Argument graph similarity · Semantic textual similarity · Text clustering

1 Introduction

As an emerging sub-field in Artificial Intelligence (AI), argumentation includes research that centers around identifying structures in natural language arguments [1]. In particular, the development of computational methods for extracting arguments and their interrelations from text [13], methods for semantic argument representation, and methods for reasoning with arguments are topics of current interest. The German Science Foundation (DFG) currently funds the special research program RATIO[1] which aims at designing robust argumenta-

[1] http://www.spp-ratio.de/home/.

C. Benzmüller and H. Stuckenschmidt (Eds.): KI 2019, LNAI 11793, pp. 101–114, 2019.
https://doi.org/10.1007/978-3-030-30179-8_8

tion machines that enable decision making and problem-solving based on arguments. ReCAP [4] as a project within RATIO focuses on the idea of argumentation machines to support researchers, decision- and policy-makers in obtaining a comprehensive overview of topic-related argumentative opinions which enable the development of a sound and credible perspective justified by convincing arguments. ReCAP aims at combining methods from case-based reasoning (CBR), information retrieval (IR), and computational argumentation (CA) to contribute to the foundations of argumentation machines. An argumentation machine can find supporting and opposing arguments for a user's topic or synthesize new arguments for an upcoming, not yet well explored topic. Thereby it could support researchers, journalists, and medical practitioners in various tasks, overcoming the limited support provided by traditional search engines used today.

This paper deals with a core problem that arises in various forms in argumentation machines, namely the clustering of arguments. Argument clustering is useful for deliberation as a method to structure a larger set of arguments dealing with a certain topic. It is also a helpful pre-processing step for the generalization and segmentation of arguments [11] to obtain reuseable patterns for case-based synthesis of new arguments. In our work we build upon arguments represented as argument graphs following the Argument Interchange Format (AIF) developed by the University of Dundee [7]. Argument graphs capture an important part of the semantics of arguments by partitioning them into claims and premises linked by various relations (rules of inference, argumentation schemes) describing how they interact. We propose a k-medoid and an agglomerative method for clustering argument graphs based on a semantic graph-based similarity measure that employs word2vec [15] as a local similarity measure to compare the textual content of claims and premises [3]. The clustering methods are evaluated using the Microtext corpus [17] and compared with pure text-based variants of the cluster methods.

The next section introduces foundations and related work. Section 3 describes the clustering algorithms, while Sect. 4 presents the experimental setup and the results obtained. The paper ends with a conclusion and possible future work.

2 Foundations and Related Work

In argumentation theory an argument consists of a set of premises and a claim together with a rule of inference which concludes from the premises to the claim. A premise can support or oppose a claim as well as an inference step. Together premises, claims, and inference steps form an argument graph. Directed graphs are suitable for formally representing the structure of the individual elements of an argument [5].

2.1 Representing Arguments as Graphs

In our work, we follow the vision of robust argumentation machines which are able to explore natural language arguments from information sources on the

web to reason with them on the knowledge level. While argument mining methods [13] aim at converting natural language argumentative texts into argument graphs containing the natural language content, our work aims at supporting the reasoning with such graphs. Thus we do not build our work upon an argumentation framework based on formal logic such Dung's argumentation framework [8] but on a graph representation that encompasses the textual content from the original natural language source of the argument. Therefore we use an argument graph representation based on the Argument Interchange Format (AIF) which was developed by the Argumentation Research Group at the University of Dundee as a standard for representing and exchanging argument graphs [7]. An argument graph can be formally defined as a 5-tuple $W = (N, E, \tau, \lambda, t)$. Here, N describes a set of nodes, $E \subseteq N \times N$ describes a set of edges, $\tau\colon N \to \mathcal{T}$ and $\lambda\colon N \to \mathcal{L}$ define functions that map nodes to types $\mathcal{T}$ and labels $\mathcal{L}$ respectively. The labels represent the textual content of a node. The set of node types $\mathcal{T}$ is declared according to the type ontology used in AIF to represent argumentation schemes. The overall topic of the argument graph is specified by a label $t \in \mathcal{L}$ [3].

Figure 1 illustrates an argument graph in AIF. On a high level perspective, claims and premises are represented as information nodes (I-nodes), depicted as grey rectangular boxes which are related to each other via scheme nodes (S-nodes), depicted as small colored rectangles. In the figure, the information node with the content *"Therefore universities should not charge fees in Germany"* without successor node represents the conclusion of this argument.

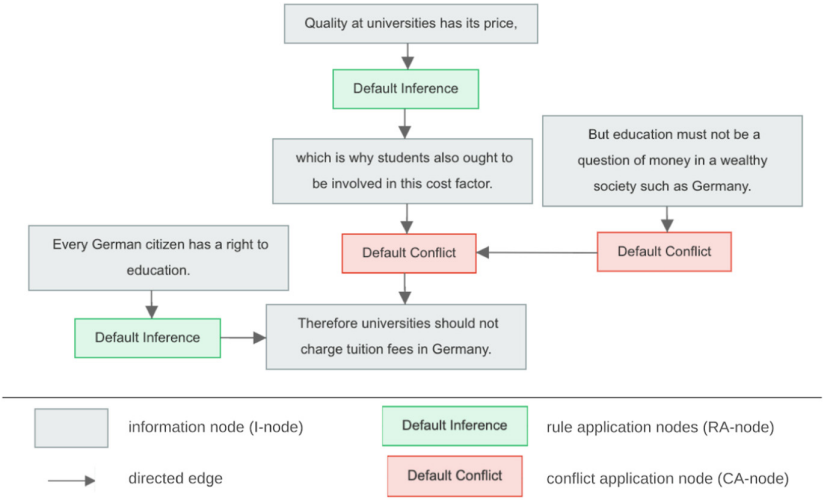

Fig. 1. Example argument graph in AIF (http://www.aifdb.org/diagram/6407.), designed with OVA (http://ova.arg-tech.org). (Color figure online)

In the simplest form applied in this work here, S-nodes can be classified into rule application nodes (RA-nodes, depicted in green color), which denote specific inference relations and conflict application nodes (CA-nodes, depicted in red color), which indicate specific conflict relations. They roughly correspond to the support and attacks relations in other frameworks. However, in general a large variety of different argumentation schemes [21] can be differentiated and represented as type of an S-node. Such argumentation schemes, correspond to archetypical forms of arguments. For example, the supporting argument can of the sub-type *Argument from Positive Consequence*, *Argument from Expert Opinion*, or *Argument from Cause to Effect*. In an other paper [12], we created an ontology consisting of 38 argumentation schemes which are arranged in a taxonomy, which is then used as a more fine grained representation of S-nodes.

2.2 Argument Clustering

In the literature, a few approaches addressing the clustering of arguments can be found. However, they make use of the pure argument text, not considering the representation of arguments in an argument graph. Boltužić and Šnajder [6] identify prominent arguments by using hierarchical clustering based on semantic textual similarity (STS). It was assumed that statements can be grouped into clusters representing abstract arguments by analyzing the degree of semantic similarity between statements. Three methods for semantic textual similarity were evaluated, including word embeddings, weighted bag-of-words and an off-the-shelf STS tool. Hierarchical agglomerative clustering was conducted using complete linkage and Ward's method. Clustering of arguments can be difficult since arguments of different topics do not always share clear boundaries. Further textual similarity might not be able to capture specific aspects of arguments.

Habernal and Gurevych [10] use a clustering of argumentative debate portal sentences and posts to derive features for argument component identification. One assumption is that a cluster contains similar arguments and cluster centroids correspond to prototypical arguments. The clustering-based features outperform other structural, syntactic and discourse features in cross-domain and cross-register evaluation.

3 Argument Graph Clustering

Clustering algorithms are used to discover similarity structures within data by grouping similar objects into clusters [14]. This paper addresses the problem of clustering argument graphs to explore structures that facilitate further reasoning through argument deliberation. Additionally, these structures can be used for the generalization and segmentation of arguments [11] to conduct case-base synthesis of new arguments. In this chapter the used base algorithms for clustering are briefly explained before their application to argument graphs is introduced. In particular the used graph-based similarity measure is described as well as various ways of deriving symmetric variants from this originally asymmetric measure to be used in the clustering algorithms.

3.1 Clustering Algorithms

Clustering algorithms aim at partitioning a data set into clusters of data points. The quality of the result can be determined by the homogeneity of the data points within a cluster and the heterogeneity to data points from other clusters. Clustering algorithms can be classified into flat and hierarchical clustering approaches. In flat clustering, the resulting clusters are without structure or relation to one another. Hierarchical clustering outputs a hierarchy among clusters. These structures are more informative and can be utilized to extract additional insights, such as cluster representatives on multiple levels of granularity. The advantage of a structured result comes at a cost of higher computational effort when compared to flat clustering.

K-medoid is a modified version of the well-known flat clustering algorithm k-means. Its objective is to divide the total quantity of data points into k disjunct clusters. K-means and k-medoid attempt to minimize the distance between data points within a cluster to the respective cluster center [14]. For the k-medoid approach the cluster center is an actual data point which is referred to as medoid. The number of clusters k must be defined prior to the process, the optimal number can be found by varying k. At the beginning of the algorithm random initial medoids are selected and the cluster distribution according to the nearest-neighbor principle is calculated. The cluster distribution is iteratively updated by improving a quality criterion. K-medoid terminates at a local optimum and therefore the cluster quality is dependent on the choice of the randomly initialized cluster centers. K-medoid is less sensitive to noisy data and errors than k-means and similar algorithms.

Agglomerative Nesting (AGNES) is a hierarchical clustering algorithm that iteratively merges previously created sub-clusters bottom-up, starting initially with a cluster for each data point [14]. In each iteration the two clusters are merged that have the smallest distance to one another. There are several alternative criteria that can be used to define closeness of two clusters and thereby the behaviour of the algorithm. In this paper we focus on the following distances measures for clusters: single linkage, complete linkage, average linkage, and Ward's method. Single linkage computes the similarity between two clusters by computing the distance of their respectively closest members. Complete linkage computes the similarity between two clusters by computing the distance of their respectively furthest members. Average linkage computes the similarity between two clusters by computing the average distance of all members. Ward's method reduces the variance within clusters by minimizing the squared error [22].

The resulting agglomerative clustering is a binary cluster tree instead of a set of disjoint clusters as in k-medoid. Although hierarchical clustering algorithms have a high space and time complexity they are suitable since we are interested in the hierarchical structure.

3.2 Similarity Measures for Clustering Argument Graphs

As we aim at clustering argument graphs, the described algorithms require a distance or similarity measure working on argument graphs. We build upon our previous work on similarity-based retrieval of argument graphs [3] in which a graph-based similarity measure for AIF graphs has been developed. This measure is used during retrieval to compare a query in the form of an argument graph with the argument graphs from a corpus.

The general principle of argument graph similarity has been adopted from process-oriented case-based reasoning [2] and follows the local-global principle [19]. The global similarity is computed from local node and edge similarities. The local node similarity $\text{sim}_N(n_q, n_c)$ of a node n_q from the query argument graph QA and a node n_c from the case argument graph CA is computed as follows:

$$\text{sim}_N(n_q, n_c) = \begin{cases} \text{sim}_I(n_q, n_c), & \text{if } \tau(n_q) = \tau(n_c) = \text{I-node} \\ \text{sim}_S(n_q, n_c), & \text{if } \tau(n_q) = \tau(n_c) = \text{S-node} \\ 0, & \text{otherwise} \end{cases}$$

The similarity of I-nodes sim_I is determined using the text contained in the I-nodes, which are typical fragments of sentences. We use a semantic textual similarity approach based on word embeddings. An embedding associates each word with a word vector in a high-dimensional real-valued vector-space. Word vectors capture the semantics of a word, in the sense that similar words have similar word vectors. We use the word2vec skip-gram [15] model to transform each word in the text of an I-node into its vector representation. The embedding vector of the whole I-node is determined by aggregating the word vectors using the weighted mean function. Prior to this, stop-words are removed. Weighting is performed using IDF weighting. Finally, the similarity sim_I is the result of the cosine measure applied to the two mean vectors.

The similarity of the S-nodes sim_S is determined comparing the types of S-nodes. It is 1, if both S-nodes are of the same type (both are RA or both are CA nodes) and 0 otherwise.

The similarity of two edges $\text{sim}_E(e_q, e_c)$ is determined based on the similarity of the nodes at their endpoints l and r respectively:

$$\text{sim}_E(e_q, e_c) = 0.5 \cdot (\text{sim}_N(e_q.l, e_c.l) + \text{sim}_N(e_q.r, e_c.r))$$

To construct a global similarity value, an admissible partial, injective mapping m is applied which maps nodes and edges from QA to CA, such that only nodes of the same type (I-nodes to I-nodes and S-nodes to S-nodes) are mapped. Edges can only be mapped if the nodes they link are mapped as well by m. For a given mapping m let sn_i be the node similarities $\text{sim}_N(n_i, m(n_i))$ and se_i the edge similarities $\text{sim}_E(e_i, m(e_i))$. The similarity for a query graph QA and a case graph CA given a mapping m is the normalized sum of the node and edge similarities (n_N is the number of nodes and n_E is the number of edges in QA).

$$\text{sim}_m(QA, CA) = \frac{sn_1 + \ldots + sn_n + se_1 + \ldots + se_m}{n_N + n_E}$$

Finally, the similarity of QA and CA is the similarity of an optimal mapping m, which can be computed using an A^* search [2].

$$\text{sim}(QA, CA) = \max_m \{\text{sim}_m(QA, CA) \mid m \text{ is admissible}\}$$

This similarity measure computes a similarity by considering the correspondence between the elements of the query argument and case argument. One particular mapping m specifies one possible correspondence which is evaluated by sim_m. In general we are interested only in the best possible correspondence, which is reflected in the overall similarity value. The graph structure of arguments enables the similarity measure to focus on the most relevant elements. When used during retrieval, it determines the best possible matching argument graph in a repository (or a case base in the terminology of case-based reasoning). Given this, it is obvious that this similarity measure is not symmetrical: query graph and case graph have clearly different roles in these measures. This can be seen easily when the query graph is a sub-graph of the case graph. In that case the query graph can be fully mapped onto the case graph and the similarity will be 1. If instead the case graph is a sub-graph of the query graph, only parts of the query graph can be mapped to the case graph and thus the similarity value will be lower than 1.

3.3 Symmetrization of Graph Similarity Measure

For clustering, a symmetric similarity measure is required to compare two argument graphs from a repository. Thus we need to make the graph similarity measure symmetric. Therefore, we propose and investigate three options in which we apply the similarity measure twice, once in each direction. We can define the symmetric graph similarity value of two argument graphs either as the minimum (1), the maximum (2), or the average (3) of the two similarity values:

$$\text{sim}_{min}(x, y) = min\{sim(x, y), sim(y, x)\} \quad (1)$$

$$\text{sim}_{max}(x, y) = max\{sim(x, y), sim(y, x)\} \quad (2)$$

$$\text{sim}_{avg}(x, y) = \frac{sim(x, y) + sim(y, x)}{2} \quad (3)$$

All symmetrization strategies are used in the evaluation. The resulting similarity measures are referred to in the following as Graph-Min, Graph-Max and Graph-Avg.

3.4 Clustering Based on Topic Vector Similarity

The proposed similarity measure has the advantage that it takes the argument structure defined by the graph into account, but it comes with the disadvantage

that it is computationally expensive due to the involved optimization problem. In order to investigate the benefit of using the graph structure over the pure textual representation of the argument, we also investigate the result of the clustering approaches with a simplified similarity measure. Therefore we employ the topic label t defined for each graph (see Sect. 2.1). This topic vector is constructed as the mean of all I-node embedding vectors, thus it is a mean value over the full argument text. Again, stop-words are removed and an IDF-weighting is applied. The topic vector similarity is again computed using the cosine similarity measure. It is already symmetric, thus the proposed symmetrization approaches are not required. Please note that the resulting clustering algorithms resemble those already investigated in the literature (see Sect. 2.2).

4 Experimental Evaluation

We now evaluate the performance of the proposed clustering methods on argument graphs. The evaluation is divided into an internal and an external evaluation. The internal evaluation has the objective to examine the quality of the clustering by using internal evaluation measures, whereas the external evaluation is supposed to determine whether the clustering is able to reconstruct the various topics reflected in a corpus.

4.1 Hypotheses

The following hypotheses are investigated in this evaluation:

- **H1:** The clustering of argument graphs is able to discover the topics contained in the clustered corpus.
- **H2:** The graph similarity measure produces a clustering which is more in line with a human classification compared to the clustering using the topic vector similarity measure.
- **H3:** A coarser topic classification is easier to reproduce by clustering than a more fine-grained classification.

4.2 Argument Corpus and Experimental Setup

As argument corpus for the clustering, an annotated corpus of argumentative microtexts by Peldzsus and Stede [17], the Microtext corpus, will be taken as basis. The corpus consists of 112 short argumentative texts about 18 different topics. One benefit of the corpus is that most texts are pre-classified into those topics. The texts were initially written in German language and then professionally translated into English. We use the English version of the corpus. All texts are annotated as argument graphs, following the scheme proposed in Peldszus and Stede [16]. The corpus is available in the AIFdb[2] in the previously explained AIF graph format (see Fig. 1 for an example). As a kind of data cleaning, a few arguments without relationship to any of the 18 topics have been removed.

[2] http://corpora.aifdb.org/.

Table 1. All topics with corresponding cluster number.

Super-Topic	Topic	# Graphs
Education	school_uniforms	3
	increase_weight_of_BA_thesis_in_final_grade	4
	charge_tuition_fees	6
Environment	waste_separation	1
	higher_dog_poo_fines	8
Medicine	health_insurance_cover_complementary_medicine	8
	over_the_counter_morning_after_pill	5
Politics	public_broadcasting_fees_on_demand	7
	stricter_regulation_of_intelligence_services	4
	introduce_capital_punishment	8
	allow_shops_to_open_on_holidays_and_Sundays	8
	buy_tax_evader_data_from_dubious_sources	2
	make_video_games_olympic	3
	EU_influence_on_political_events_in_Ukraine	3
Living	cap_rent_increases	6
	keep_retirement_at_63	6
	partial_housing_development_at_Tempelhofer_Feld	2
	TXL_airport_remain_operational_after_BER_opening	3

To evaluate the effect of coarser-grained topics on the clustering quality, we manually combined the topics to five classes, or super-topics. The goal is to reduce the false classification of graphs with similar topics, for instance, the topics "school uniforms" and "charge tuition fees" were merged into the more general class "education". The generated topic groups are about the same size with the exception of "politics", which is the largest cluster. Table 1 shows all topics of the corpus together with the number of argument graphs and the super-topic. As word representation we used the pre-trained Google News[3] word embeddings for English language.

4.3 Internal Evaluation

In the internal evaluation the quality of the clustering will be evaluated by three measures. The silhouette coefficient utilizes both cohesion and separation into one value from $[-1, 1]$ [20]. Cohesion measures how closely objects in a cluster are related to each other and separation how distinct objects from different clusters are. A high value indicates that objects are well matched within clusters and poorly between clusters whereas a low value corresponds to the opposite. The Dunn index (DI) is a ratio of within cluster and between cluster separations [9].

[3] https://code.google.com/archive/p/word2vec/.

DI has values in $[0, \infty]$ and should be maximized as well. Connectivity measures to what extent items are placed in the same cluster as their nearest neighbor with values in $[0, \infty]$. Unlike the other measures lower values correspond to a better clustering quality. For each measure the optimal k is found, which equates the highest value of the respective measure. Evaluation was performed with a varying number k, which ranged from 2 to 18.

Table 2 displays the results for k-medoid clustering using the different similarity measures. In this and the subsequent tables, the best values for the measures are marked in bold font. The best silhouette coefficient is achieved with two clusters. It is notable that topic vectors produce 18 clusters which corresponds to the number of topics in the corpus. Nevertheless, the silhouette coefficient is clearly lower than for the graph similarity, where the value varies from 0.29 to 0.38. This can be interpreted as a weak cluster structure with high separation and low cohesion [11]. DI values are comparable and the number of clusters is relatively high, ranging from 5 to 18 clusters. The connectivity is comparatively low, which is due to the small number of clusters. Topic vectors produce the lowest connectivity score.

Table 2. Cluster results for k-medoid.

	Graph-Min	Graph-Max	Graph-Avg	Topic vectors
Silhouette	**0.38**	0.29	0.35	0.11
Optimal k	2	2	2	18
Dunn index	0.52	0.59	**0.60**	0.57
Optimal k	15	18	5	16
Connectivity	2.96	4.59	7.59	**2.90**
Optimal k	2	3	2	2

The results for AGNES with different linkage methods (SL = single linkage, CL = complete linkage, AL = average linkage, W = Ward's method) are shown in Table 3. The results are quite comparable to the k-medoid clustering, also with regard to the number of clusters. DI comes with a higher k, whereas the silhouette coefficient and connectivity have mostly $k = 2$. The silhouette coefficient varies for the graph similarity from 0.26 to 0.39 and the values for the clustering with topic vectors are much smaller (0.10 to 0.16). The connectivity and DI values are very similar to the values in Table 2 with topic vectors having the overall best values for connectivity.

The internal measures show a weak cluster structure for both similarity methods which could be artificial. For silhouette coefficient and connectivity in most of the configurations the optimal number of clusters is two. However, DI and topic vector cluster analysis for silhouette coefficient prefers a higher number of clusters in k-medoid and AGNES. In comparison, the graph-based similarity measures produce slightly better results for the internal measures.

Table 3. Cluster results for AGNES.

	Graph-Min				Graph-Max				Graph-Avg				Topic vectors			
	SL	CL	AL	W	SL	CL	AL	W	SL	CL	AL	W	SL	CL	AL	W
Silhouette	**0.39**	0.29	0.36	0.38	0.31	**0.32**	**0.32**	0.26	0.34	0.35	0.35	**0.36**	0.10	0.11	**0.16**	0.12
Optimal k	2	4	2	2	2	2	2	3	2	3	3	2	2	16	2	5
Dunn index	0.52	**0.64**	0.59	**0.64**	**0.72**	**0.72**	**0.72**	0.66	0.53	**0.66**	**0.66**	0.61	0.62	**0.64**	0.60	0.61
Optimal k	9	18	15	18	18	14	14	18	10	17	17	17	5	18	17	16
Connectivity	3.27	12.14	4.39	**2.96**	**2.93**	11.18	5.14	5.88	4.73	4.73	4.73	**2.90**	2.93	10.76	3.03	**2.90**
Optimal k	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

4.4 External Evaluation

The external evaluation focuses on the classification of each graph in relation to their respective topic as shown in Table 1. It is supposed to show whether the clustering complies with the topics and is applicable for further use within the envisioned argumentation machine.

As measures, precision and recall are calculated and combined in the Rand index (RI). RI compares the manual classification (M) and the clustering (C) based on the agreement and disagreement between object pairs in both partitionings [18]. Precision and recall are calculated based on the number of common pairs in both sets (true positives), the number of pairs in C but not in the M (false positives), vice versa (false negatives) and the number of different pairs in both sets (true negatives). For further analysis the resulting classification accuracy is calculated for the clustering with the highest RI.

Table 4 displays the RI values with k equal to the number of classes in the corpus. It shows that all methods achieve very good results, whereas the topic vectors deliver the best value for all configurations. Only AGNES with single linkage leads to significantly lower values.

Table 4. Rand index results $k = 18$.

	k-medoid	AGNES			
		SL	CL	AL	W
Graph-Min	0.86	0.67	0.87	0.82	0.85
Graph-Max	0.87	**0.83**	0.86	0.84	0.87
Graph-Avg	0.89	0.69	0.88	0.85	0.89
Topic vectors	**0.97**	0.60	**0.96**	**0.92**	**0.97**

In order to gain a more detailed insight we evaluated the quality of the clustering for k-medoid with topic vector similarity (k = 18). 11 of 18 topics are perfectly in accordance with the classification, no graph is assigned to a different cluster. Only for two topics no graph is classified correctly, although this affects only three graphs. Furthermore, wrongly classified graphs are often

assigned to a cluster with similar content, for example “waste separation” and “higher dog poo fines”. 74 of 87 graphs are classified correctly, which leads to an overall classification accuracy of 85%. Hypothesis H1 can therefore be accepted. The graph similarity methods classify only about 25 graphs correctly with a classification accuracy of 29%. The best results are achieved with the topic vector representations and not the graph representations. Hypothesis H2 needs thus to be rejected. This also implies that a high RI does not automatically lead to a good classification. When looking at the parameters of the RI, the pairs of true positives only have a small influence. Thus the reason for the high RI values are true negatives, which have a strong impact on the equation.

Table 5 displays the RI values for the five super-topics. The best result is achieved with topic vectors and k-medoid clustering. AGNES with SL produces very low RI scores. All values are clearly worse than the ones from Table 4. Thus it is shown that cluster analysis with a smaller number of clusters does not automatically lead to better classification.

Table 5. Rand index results for $k = 5$.

	k-medoid	AGNES			
		SL	CL	AL	W
Graph-Min	0.64	0.30	0.59	0.57	0.62
Graph-Max	0.66	**0.34**	0.58	0.49	0.64
Graph-Avg	0.65	0.32	**0.61**	**0.62**	0.62
Topic vectors	**0.82**	0.31	0.60	0.30	**0.66**

The overall classification accuracy of k-medoid with topic vectors is 68%. In comparison to topics, the classification accuracy for the super-topics is lower and only 59 of 87 graphs are correctly clustered instead of 74. The classification for education and environment is perfect. For medicine, the value is nearly perfect (92%). However, the accuracy decreases to 71% or even 0% for the last two super-topics. Negative outlier is the first cluster, where two topics are covered.

Despite this degradation, the topic vector representation still produces the best result. The best graph similarity using max symmetrization classifies only 25 graphs correctly which leads to an accuracy of 28%. This emphasizes the impression that the graph-based similarity measures are not suited for thematic clustering, since the resulting clusters contain argument graphs of different topics. Overall, the results for the more fine-grained topic classification are better for both, RI and classification accuracy. Thus, hypothesis H3 also needs to be rejected.

5 Conclusion

In this paper, we proposed two clustering algorithms for argument graphs which are based on a graph-based similarity measure. The motivation behind this is the

hypothesis that the graph structure of the argument provides relevant semantic information about the argument and thus should enable a better clustering as when working on the pure text only. The results, however, show that clustering using the graph-based similarity is only slightly better in the internal evaluation measures, while the clustering based on the topic vector similarity outperforms the graph-based approach in identifying the topics of the arguments. Thus it became obvious that for pure topic discovery the structure of the argument is not really relevant as the words occurring in the text already allow to identify the topic appropriately. Here, the use of the structure seems to distract the clustering process.

However, argument graph clustering shall not only be used to group arguments w.r.t. their topic. In particular for the purpose of generalization, building clusters of graphs with similar structure but different topic is more important as the generalization over different topics is desirable. Further investigation is required to find out whether the proposed clustering approach using the graph-based measure is appropriate for this purpose.

Further, it should be noted that the current evaluation is only performed using a quite small corpus. Thus, more extensive evaluations are necessary, in particular using argument graph corpora automatically mined from text, allowing a larger number of graphs to be used.

During the course of our future work towards argument synthesis with case-based reasoning involving adaptation methods, clustering will play a pivotal role to infer a structure on the case base. Only in the context of this application it will become clear whether the proposed algorithms are able to produce useful clusters.

Acknowledgments. This work was funded by the German Research Foundation (DFG), project 375342983.

References

1. Atkinson, K., et al.: Towards artificial argumentation. AI Mag. **38**(3), 25–36 (2017)
2. Bergmann, R., Gil, Y.: Similarity assessment and efficient retrieval of semantic workflows. Inf. Syst. **40**, 115–127 (2014). https://doi.org/10.1016/j.is.2012.07.005
3. Bergmann, R., Lenz, M., Ollinger, S., Pfister, M.: Similarity measures for case-based retrieval of natural language argument graphs in argumentation machines. In: Proceedings of the 32nd International Florida Artificial Intelligence Research Society Conference, FLAIRS 2019, Sarasota, Florida, USA. AAAI-Press (2019)
4. Bergmann, R., Schenkel, R., Dumani, L., Ollinger, S.: ReCAP - information retrieval and case-based reasoning for robust deliberation and synthesis of arguments in the political discourse. In: Proceedings of the Conference "Lernen, Wissen, Daten, Analysen", LWDA 2018, 22–24 August 2018, Mannheim, Germany, CEUR Workshop Proceedings, vol. 2191. CEUR-WS.org (2018). http://ceur-ws.org/Vol-2191/paper6.pdf
5. Bex, F., Reed, C.: Schemes of inference, conflict and preference in a computational model of argument. Stud. Logic Grammar Rhetoric **23**(36), 39–58 (2011)

6. Boltužić, F., Šnajder, J.: Identifying prominent arguments in online debates using semantic textual similarity. In: Proceedings of the 2nd Workshop on Argumentation Mining, pp. 110–115. Association for Computational Linguistics, June 2015
7. Chesnevar, C., et al.: Towards an argument interchange format. Knowl. Eng. Rev. **21**(4), 293–316 (2006)
8. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. **77**(2), 321–358 (1995)
9. Dunn, J.C.: A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J. Cybern. **3**(3), 32–57 (1973). https://doi.org/10.1080/01969727308546046
10. Habernal, I., Gurevych, I.: Exploiting debate portals for semi-supervised argumentation mining in user-generated web discourse. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2127–2137 (2015)
11. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data - An Introduction to Cluster Analysis. Wiley, New York (1990)
12. Lenz, M., Ollinger, S., Sahitaj, P., Bergmann, R.: Semantic textual similarity measures for case-based retrieval of argument graphs. In: Case-Based Reasoning Research and Development: 27th International Conference, ICCBR 2019, 8–12 September 2019, Otzenhausen, Germany, Proceedings. Springer, Heidelberg (2019, accepted for publication)
13. Lippi, M., Torroni, P.: Argument mining from speech: detecting claims in political debates. In: Schuurmans, D., Wellman, M.P. (eds.) Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016), pp. 2979–2985. AAAI Press (2016). http://argumentationmining.disi.unibo.it/publications/AAAI2016.pdf
14. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, New York (2008)
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781 [cs], January 2013
16. Peldszus, A., Stede, M.: From argument diagrams to argumentation mining in texts. Int. J. Cogn. Inform. Nat. Intell. **7**(1), 1–31 (2013). https://doi.org/10.4018/jcini.2013010101
17. Peldszus, A., Stede, M.: An annotated corpus of argumentative microtexts. In: First European Conference on Argumentation: Argumentation and Reasoned Action, Portugal, Lisbon, June 2015. http://www.ling.uni-potsdam.de/~peldszus/eca2015-preprint.pdf
18. Rand, W.: Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. **66**(336), 846–850 (1971)
19. Richter, M.M., Weber, R.O.: Case-Based Reasoning - A Textbook. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40167-1
20. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**(1), 53–65 (1987). https://doi.org/10.1016/0377-0427(87)90125-7
21. Walton, D., Reed, C., Macagno, F.: Argumentation Schemes. Cambridge University Press, Cambridge (2008)
22. Ward, J.H.: Hierarchical grouping to optimize an objective function. J. Am. Stat. Assoc. **58**(301), 236–244 (1963). http://www.jstor.org/stable/2282967