# The ReCAP Project

## Similarity Methods for Finding Arguments and Argument Graphs

Ralph Bergmann[1] · Manuel Biertz[1] · Lorik Dumani[1] · Mirko Lenz[1] · Anna-Katharina Ludwig[1] ·
Patrick J. Neumann[1] · Stefan Ollinger[1] · Premtim Sahitaj[1] · Ralf Schenkel[1] · Alex Witry[1]

**Abstract**
Argumentation Machines search for arguments in natural language from information sources on the Web and reason with them on the knowledge level to actively support the deliberation and synthesis of arguments for a particular user query. The RECAP project is part of the Priority Program RATIO and aims at novel contributions to and confluence of methods from information retrieval, knowledge representation, as well as case-based reasoning for the development of future argumentation machines. In this paper we summarise recent research results from the project. In particular, a new German corpus of 100 semantically annotated argument graphs from the domain of education politics has been created and is made available to the argumentation research community. Further, we discuss a comprehensive investigation in finding arguments and argument graphs. We introduce a probabilistic ranking framework for argument retrieval, i.e. for finding good premises for a designated claim. For finding argument graphs, we developed methods for case-based argument retrieval considering the graph structure of an argument together with textual and ontology-based similarity measures applied to claims, premises, and argument schemes.

## 1 Introduction

Argumentation is ubiquitous and a fundamental part of our lives. People use arguments to inform themselves or to form opinions, or to convince others towards a certain standpoint. The Web offers plenty of arguments on many topics, but due to its size it is almost impossible for humans to find all arguments on a topic in a reasonable amount of time. Not finding all relevant arguments on a sensitive topic may lead to a biased view and consequently to bad decisions.

The RECAP project is part of the RATIO priority program[1] and aims at the vision of future argumentation machines.

---

✉ Lorik Dumani
dumani@uni-trier.de

[1] Trier University, Trier, Germany

On the application side, the project focuses on political scientists, journalists, and human decision makers and aims to support them in obtaining an overview of current arguments on a specific topic and in forming personal opinions based on convincing arguments. Contrary to existing search engines, which primarily operate on the textual level, such argumentation machines will reason on the knowledge level formed by argumentative propositions and argumentation structures. In this context, our aim is to develop methods that are able to capture arguments in a robust and scalable manner, in particular representing, contextualising, aggregating, and synthesising arguments and making them available to users.

This paper summarises the results we accomplished in the project so far. When we talk about an *argument* we mean a combination of a *claim* (or conclusion) and several *premises* (or reasons) together with one or several inference rules linking them [22]. Claims and premises are also called *Argumentative Discourse Units* (ADU) [22] and the inference rules between them are also called *argument schemes*. Walton [30] comprehensively describes typical argument schemes which occur in natural language argumentation.
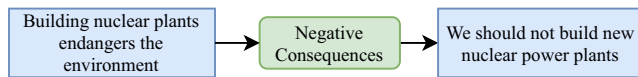
**Fig. 1** A simple argument graph showing the argument scheme "Negative Consequences" by Walton [30] for the inference from premise to claim

Arguments are represented as *argument graphs* with nodes representing ADUs and argument schemes and edges representing their relationships. Fig. 1 visualises a simple argument graph.

Next, we present our view of the architecture of an argumentation machine. In Sect. 3, we introduce a new corpus for evaluating the methods developed in the project. Sect. 4 discusses the proposed methods and their evaluation. Finally, Sect. 5 concludes the paper and presents our directions for future work.

## 2 Architecture of an Argumentation Machine

We now outline the envisioned architecture of our argumentation machine (see Bergmann et al. [5] for further details). Fig. 2 illustrates this architecture and shows the different research fields and their interrelations.

The bottom part of this layered architecture shows the textual level of the argumentation machine. It addresses argument mining as well as corpus construction from existing textual sources, leading to semantically annotated argumentation graphs that reflect the content of documents on the knowledge level. Note that the argumentation machine works closely with argumentation structures in natural language, but in order to achieve argumentative reasoning, it abstracts from the raw text by using similarity measures, fact extraction, validation, clustering, generalisation, and adaptation of arguments, thereby offering some form of argument competency. With the term *similarity* we refer to both the similarity of two ADU nodes, e.g. measured by textual similarity, and to the similarity of two graphs, by considering also structural aspects. *Retrieval* addresses
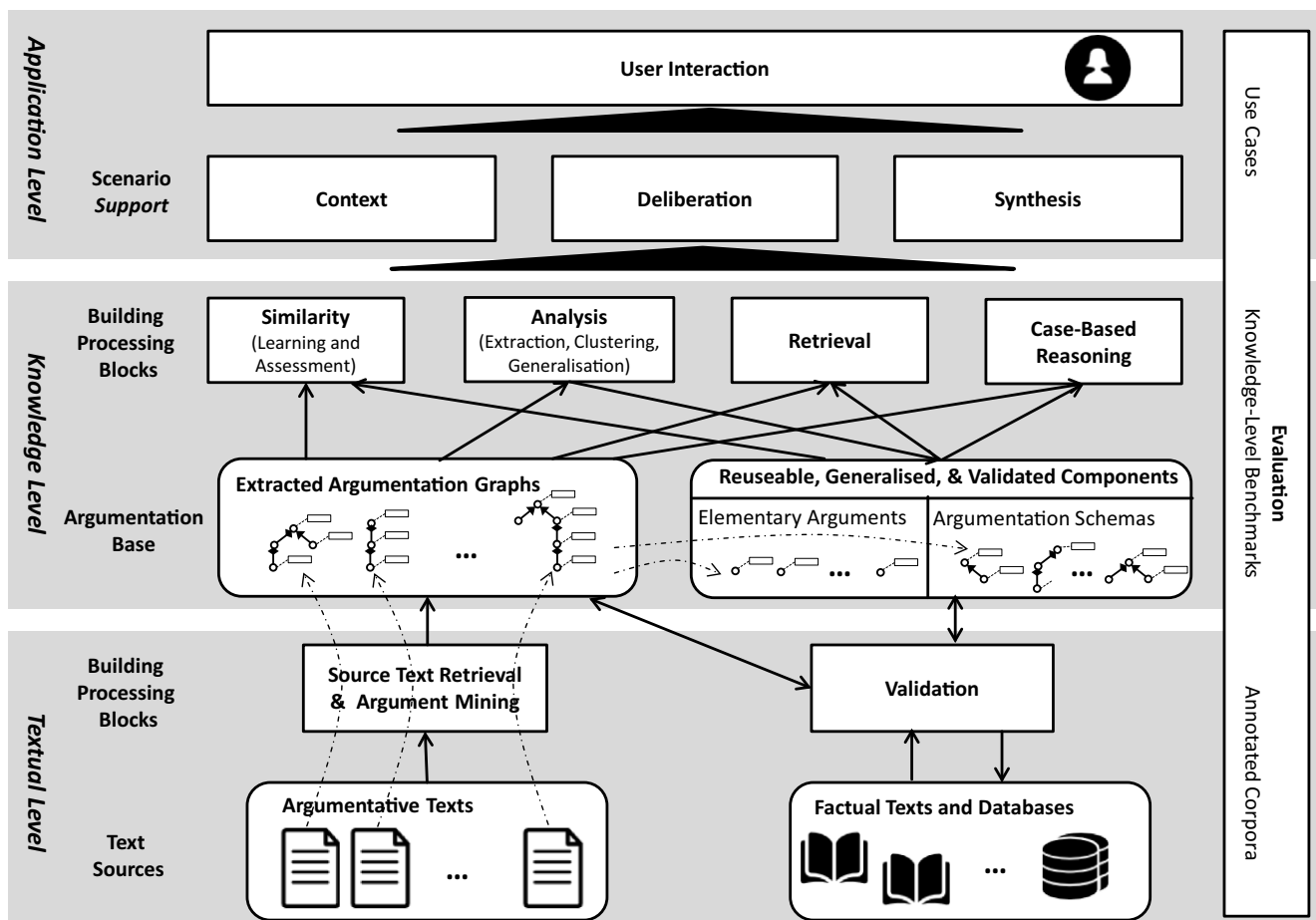


**Fig. 2** Architecture of the argumentation machine

the finding and ranking of argument nodes and argument graphs in terms of their relevance and factual correctness. *Validation* of facts can be done, for instance, by querying the information in knowledge graphs or by reformulating a fact as a search query on the Web. *Case-based reasoning* allows analogical reasoning to transfer an argument graph to a new context.

The application level allows the development of *deliberation* and *synthesis* applications using the methods from the knowledge level. For example, applications can support finding and weighting all arguments supporting or opposing some claim, based on the available knowledge. Applications can also try to generate new arguments for an upcoming topic by transfer and combination of existing relevant arguments from a closely related topic. The context module aims at capturing, analysing, and representing the specific user's context, i.e. the specific issue under consideration as well as specific beliefs and constraints of the user.

## 3 Building a High Quality Corpus

In a requirements acquisition workshop with experts from the fields of journalistic writing and political research we elaborated concrete use cases for the envisioned argumentation machine. These use cases guide our methodological research and will serve in the future to build selected applications for deliberation and synthesis. We have chosen the topic of education policy, as it is relevant to society, moderately complex, and relatively easy to understand. In particular, education policy varies from federal state to federal state in Germany, but related issues are discussed throughout the country. Thus, we expect that although this field covers a rich spectrum of topics, the transfer of arguments from one state to another could be investigated.

As no corpus of argument graphs on education policy in Germany was available, we developed a new corpus consisting of arguments from the political discourse in the three federal states Rhineland-Palatinate, Hamburg, and Bavaria [12]. Since argument mining methods are still under development and currently do not produce semantically annotated argument graphs of sufficient quality, we created the corpus manually. For this purpose, we selected texts from high-quality sources such as press releases, newspaper commentaries, and election programs. The argumentative contents were independently annotated and converted into argument graphs by two annotators using a modified variant[2] of the OVA tool [16]. During the construction of the graphs, the argument schemes proposed by Walton et al. [30] were used, which enable a very detailed representation of the different types of inferences occurring in the

documents. In weekly discussions the two graphs per text source were merged into one gold standard. The resulting corpus consists of 100 argument graphs, with about 25 nodes and 20 edges in average per graph. It is available to the argument mining community on request.

As the overall construction and validation of the corpus took about 18 months, we also considered existing corpora during the development of the proposed methods. This includes the Potsdam Argumentative Microtext corpus [23] that is available in German and English. However, as it only includes inferences annotated with support or attack relations, we refined the annotations using appropriate argument schemes. In addition we crawled debate portals such as idebate.org and debatewise.com to create corpora of claims with premises supporting or attacking them.

## 4 Retrieval and Case-Based Reasoning with Arguments and Argument Graphs

We now present selected approaches for retrieval and reasoning with arguments from the knowledge level of the architecture.

### 4.1 Matching Similar Claims by Textual Similarity

In an initial study [14], we evaluated different methods for claim similarity. We built upon the groundwork of Wachsmuth et al. [29], who set up an argument search engine based on crawling and indexing arguments from four debate portals. Since their corpus was not freely available at that time, we built a comparable corpus with 63,250 claims and about 695,000 premises by crawling the same portals. For our evaluation we used 232 claims from this corpus on the topic *energy*. To determine these claims, we first identified the 44 most similar words to *energy* using a pre-trained word2Vec [21] model, and then randomly chose 232 *query claims* amongst all claims containing at least one of them. We then evaluated how well 196 text similarity methods implemented in Apache Lucene performed in finding relevant *result claims* for these query claims. To build a gold standard, we constructed a result pool for each query from the top five results of each method, resulting in a total of 3,622 (query, result) pairs. Each pair was then assessed by two annotators on a scale from 1 (semantically dissimilar) to 5 (semantically equal). For each method, the result quality was then measured using the established nDCG metric [17]. Our results show that the widely used BM25 method [26] performs very well with an nDCG@5 of 0.7944, but an even better performance (0.8355) was achieved by a combination of Axiomatic Approaches for IR and Divergence from Randomness (DFR) [1]. The results of our experiments also support the intuitive assumption

---

[2] http://ova.uni-trier.de/.

that, given a query claim, the premises of a similar claim are more relevant to the query claim than those of a dissimilar one, using a second set of relevance assessments for (query claim, result premise) pairs on a binary scale.

## 4.2 A Probabilistic Ranking Framework for Argument Retrieval

For finding good premises for a query claim from a large corpus of already mined arguments, we proposed a principled probabilistic ranking framework [13]. Given a controversial claim or topic, the system first identifies highly similar claims in the corpus, and then clusters and ranks their supporting and attacking premises, taking clusters of claims as well as the stances of query and premises into account.

The description of the whole framework is beyond the scope of this paper. We only sketch the approach for finding supporting premises to a query claim; finding attacking premises is analogous. Given a large corpus of claims and premises, we first create a set of disjoint claim clusters $\Gamma = \{\gamma_1, \gamma_2, ...\}$ where each cluster $\gamma_j$ consists of claims with the same meaning. Analogously, we create a set of disjoint premise clusters $\Pi = \{\pi_1, \pi_2, ...\}$ consisting of premises with the same meaning. Our goal is to find the best clusters of supporting premises $\pi^+$ for a query $q$. To do so, we estimate the probability of relevance $P(\pi^+|q)$ for each $\pi^+ \in \Pi$. This probability is high if many premises from the cluster strongly support claims relevant to the query claim. To quantify this, we consider the probability $P(c|q)$ that claim $c$ is relevant for query $q$ and the probability $P(p^+|c, q)$ that a user would pick premise $p$ amongst all supporting premises of $c$. We then obtain $P(p^+|q)$ by adding $P(c|q) \cdot P(p^+|c, q)$ over all claims in the corpus, and can compute $P(\pi_j^+|q)$ as the sum of $P(p'^+|q)$ over all premises $p'^+ \in \pi_j^+$.

We can estimate $P(c|q)$ with standard text retrieval methods; in our experiments, we use DFR, the best method for claim retrieval (see Sect. 4.1). Regarding premises, we prefer premises that appear often within a claim cluster but disfavour premises that appear within most or even all claim clusters; this is the same principle used in the tf-idf weight [27]. We thus estimate $P(p^+|c)$ as the product of two frequency statistics (plus normalisation): the *premise frequency* $pf(p^+, c)$, i.e. the frequency with which $p$ is used as support for claims equivalent to $c$ (i.e. within $c$'s claim cluster), and the *inverse claim frequency* $icf(p^+)$, i.e. the inverse number of claim clusters for which $p$ is used as support.

We evaluated our ranking framework using the dataset introduced in Sect. 4.1. We calculated all claims' and premises' embeddings utilising BERT [11]. We then clustered the claims in an offline operation with agglomerative clustering [15] and obtained clusters by applying a dynamic tree cut [18]. Premise clusters relevant to the query are determined with the same method at query time, considering the premises of the claims most similar to the query and the ten most similar premises to each of these premises determined by BM25. We randomly picked 30 query claims out of the 232 claims. As a baseline system, we implemented the approach proposed by Wachsmuth et al. [29]. Two annotators assessed the 1,195 premises retrieved by at least one system on a three-fold relevance scale. Our approach significantly outperformed the baseline for nDCG@5.

## 4.3 Case-Based Reasoning for Retrieval and Adaptation of Argument Graphs

Besides methods from information retrieval we also investigated case-based reasoning (CBR) methods [2, 25] applied to cases in the form of argument graphs. CBR is a method from knowledge-based problem solving based on experiential knowledge, called cases. It allows the retrieval of cases similar to a query but also the adaptation of cases towards the query. Thus, retrieval methods from CBR can be used as an alternative approach to information retrieval and they are particularly useful for whole argument graphs as their argumentative structure can be considered during similarity assessment. Further, adaptation methods from CBR can be applied to the adaptation of argument graphs. Both issues are subject of investigation in the project.

In our work [4, 20] we aim at retrieving and adapting argument graphs from a repository (called *case-base* in CBR terminology). Formally, an argument graph is a semantically labeled directed graph and represented as a tuple $A = (N, E, \tau, \lambda, t)$ [3]. $N$ is the set of nodes and $E \subseteq N \times N$ is the set of directed edges connecting two nodes. $\tau : N \to \mathcal{T}$ assigns each node a type and $\lambda : N \to \mathcal{L}$ assigns each node a semantic description from a language $\mathcal{L}$. $t \in \mathcal{L}$ describes the overall topic of the argument represented in the graph. The types $\mathcal{T}$ follow the AIF standard [9] so that a node can either be an I-node with natural language propositional content or an S-node characterized by the respective argumentation scheme. The mapping function $\lambda$ is used to link a semantic representation to a node. For an I-node $n$, $\lambda(n)$ is the original textual representation (possibly after traditional pre-processing such as stopword removal) together with a semantic representation of this text in the form of a vector, produced by a sentence encoder.

A query to be used in retrieval is also an argument graph or a partial argument graph, which can consist of one or a few (maybe linked) nodes only. For example, a claim with a few premises can be used as a query to retrieve a set of graphs that contribute additional premises for the claim or other sub-graphs supporting or attacking the premises in the query.

For case retrieval, a graph-based similarity measure has been developed which allows to assess the similarity between a query graph $QA$ and a case graph $CA$ form the repository. The graph similarity is computed based on a local node similarity measure $sim_N(n_q, n_c)$ of a node $n_q$ from the query argument graph $QA$ and a node $n_c$ from the case argument graph $CA$ and an edge similarity measure $sim_E(e_q, e_c) = 0.5 \cdot (sim_N(e_q.l, e_c.l) + sim_N(e_q.r, e_c.r))$ which assesses the similarity of an edge $e_q$ from $QA$ and an edge $e_c$ from $CA$.

To construct a global graph similarity value, an admissible 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 $sim_N(n_i, m(n_i))$ and $se_i$ the edge similarities $sim_E(e_i, m(e_i))$. The similarity for a query graph $QA$ and a case graph $CA$ given a mapping $m$ is the normalised sum of the node and edge similarities: $sim_m(QA, CA) = (sn_1 + \cdots + sn_n + se_1 + \cdots + se_m)/(n_N + n_E)$ Finally, the similarity of $QA$ and $CA$ is the similarity of an optimal mapping $m$, which can be computed using an $A^*$ search [3], i.e., $sim(QA, CA) = \max_m\{sim_m(QA, CA) \mid m \text{ is admissible}\}$

For similarity-based retrieval of argument graphs from a case base, a linear retrieval approach should be avoided due to unacceptable retrieval times caused by the complexity of $A^*$ search as well as the complexity of the involved node similarity measures. Thus, we applied a two-phase approach, which divides the retrieval into an efficient pre-filter stage followed by phase in which only the filtered cases are assessed in depth using the complex graph similarity measure. We implemented the pre-filter as a linear similarity-based retrieval of the cases based only on the semantic similarity of the topic vector $t$ [4]. The filter selects the $k$ most similar cases, which are passed over to the second phase which implements the ranking by a linear assessment of the cases using the graph-based similarity as described above.

This approach significantly depends on the methods used to assess the similarity of nodes. For S-nodes representing argument schemes their similarity is determined according to the closeness of the schemes within a taxonomic ontology of argument schemes [20]. Therefore, we apply a similarity measure proposed by Wu and Palmer [31] that considers the depth of the two schemes to be compared and the length of the taxonomy path to their closest common predecessor. For I-nodes, their textual information can be compared by textual similarity measures. In order to capture the semantic closeness of the I-nodes, we investigated various word and sentence embedding methods assessing the similarity.

In a first paper [4], we used plain word2vec Skip-gram embeddings (WV) [21] applied to the pre-processed node text (tokenisation and an optional stopword removal). The similarity between two I-nodes is then assessed using the cosine similarity applied to the aggregated embedding vectors of the words in the pre-processed text. We further extend this investigation by considering various alternative embedding approaches [20] as well as combinations of them with alternative vector similarity measures. In particular, the unsupervised methods fastText [7] and GloVe [24] (word embeddings) as well as the distributed memory model of paragraph vectors (DV) [19] (sentence embedding) have been applied. In addition, the supervised sentence embedding methods InferSent [10] (based on BiL-STMs) and the Universal Sentence Encoder [8] variants USE-T and USE-D have been investigated as well as various combinations based on vector concatenation. In experiments using the semantically extended Potsdam Microtexts Corpus [23], the USE-T achieved the highest Average Precision of 0.972 whereas WV achieved the highest nDCG@10 of 0.877.

Besides their use in retrieval, we also investigated the use of the argument graph similarity measures for clustering the argument graphs in the repository w.r.t. their similarity [6]. Clusters of graphs can then be used for further research on generalisation of graphs as pre-processing step for argument graph adaptation. In addition, we approach argument graph adaptation by analogical reasoning. For this purpose, we further enhance the argument graph representation by identifying noun chunks in the text of the I-nodes and linking them to concepts in the ConceptNet knowledge graph [28] as a means to represent background knowledge. Based on the knowledge graph, various substitutions of the concepts can be performed as a means for argument adaptation. For example, generalisations can be determined which can be further specialised differently towards the concepts in the query node. Also shortest paths in the knowledge graph between the core concepts occurring in the I-nodes of an argument graph can be determined as a source for analogical transfer to different concepts occurring in the query. Respective methods are currently being implemented and tested.

## 5 Conclusion and Future Work

This paper summarised the first results of the RECAP project. We created a corpus of 100 high-quality graphs in German language on which we and the argument community can develop and evaluate argument mining methods. Apart from that, we implemented and evaluated methods for finding the best arguments and argument graphs on existing corpora. Future work will elaborate the methods for argument adaptation. Further comprehensive evaluations

based on the elaborated use cases and the developed corpus will be performed.

## References

1. Amati G, van Rijsbergen CJ (2002) Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Trans Inf Syst 20(4):357–389. https://doi.org/10.1145/582415.582416
2. Bergmann R (2002) Experience management: Foundations, development methodology, and Internet-based applications. Springer, Heidelberg, Berlin, New York https://doi.org/10.1007/3-540-45759-3
3. Bergmann R, Gil Y (2014) Similarity assessment and efficient retrieval of semantic workflows. Inf Syst 40:115–127. https://doi.org/10.1016/j.is.2012.07.005
4. Bergmann R, Lenz M, Ollinger S, Pfister M (2019) Similarity measures for case-based retrieval of natural language argument graphs in argumentation machines. In: FLAIRS. AAAI Press, Palo Alto, pp 329–334
5. Bergmann R, Schenkel R, Dumani L, Ollinger S (2018) Recap – information retrieval and case-based reasoning for robust deliberation and synthesis of arguments in the political discourse. In: LWDA. CEUR workshop proceedings, vol 2191, pp 49–60
6. Block K, Trumm S, Sahitaj P, Ollinger S, Bergmann R (2019) Clustering of argument graphs using semantic similarity measures. In: KI. Springer, Heidelberg, Berlin, New York, pp 101–114 https://doi.org/10.1007/978-3-030-30179-8_8
7. Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Linguist 5:135–146. https://doi.org/10.1162/tacl_a_00051
8. Cer D, Yang Y, Kong SY, Hua N, Limtiaco N, John RS, Constant N, Guajardo-Cespedes M, Yuan S, Tar C, Sung YH, Strope B, Kurzweil R (2018) Universal sentence encoder (http://arxiv.org/abs/1803.11175)
9. Chesñevar CI, McGinnis J, Modgil S, Rahwan I, Reed C, Simari GR, South M, Vreeswijk G, Willmott S (2006) Towards an argument interchange format. Knowl Eng Rev 21(4):293–316. https://doi.org/10.1017/S0269888906001044
10. Conneau A, Kiela D, Schwenk H, Barrault L, Bordes A (2017) Supervised learning of universal sentence representations from natural language inference data, pp 670–680 https://doi.org/10.18653/v1/D17-1070
11. Devlin J, Chang M, Lee K, Toutanova K (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT, pp 4171–4186
12. Dumani L, Biertz M, Witry A, Ludwig AK, Lenz M, Ollinger S, Bergmann R, Schenkel R (2020) The ReCAP corpus: A corpus of complex argument graphs on german education politics. https://www.uni-trier.de/fileadmin/fb4/prof/INF/DBI/Publikationen/preprint_dumani_corpus.pdf. Accessed 06th June 2020
13. Dumani L, Neumann PJ, Schenkel R (2020) A framework for argument retrieval. In: ECIR, pp 431–445 https://doi.org/10.1007/978-3-030-45439-5_29
14. Dumani L, Schenkel R (2019) A systematic comparison of methods for finding good premises for claims. In: SIGIR, pp 957–960 https://doi.org/10.1145/3331184.3331282
15. Jain AK, Dubes RC (1988) Algorithms for clustering data. Prentice-Hall, Upper Saddle River
16. Janier M, Lawrence J, Reed C (2014) OVA+: an argument analysis interface. In: COMMA, pp 463–464 https://doi.org/10.3233/978-1-61499-436-7-463
17. Järvelin K, Kekäläinen J (2002) Cumulated gain-based evaluation of IR techniques. ACM Trans Inf Syst 20(4):422–446. https://doi.org/10.1145/582415.582418
18. Langfelder P, Zhang B, Horvath S (2009) Dynamic tree cut: In-depth description, tests and applications
19. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: ICML. JMLR workshop and conference proceedings, vol 32, pp 1188–1196 (JMLR.org)
20. Lenz M, Ollinger S, Sahitaj P, Bergmann R (2019) Semantic textual similarity measures for case-based retrieval of argument graphs. In: ICCBR. Springer, Heidelberg, Berlin, New York, pp 219–234 https://doi.org/10.1007/978-3-030-29249-2_15
21. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. In: NIPS, pp 3111–3119
22. Peldszus A, Stede M (2013) From argument diagrams to argumentation mining in texts. IJCINI 7(1):1–31. https://doi.org/10.4018/jcini.2013010101
23. Peldszus A, Stede M (2015) An annotated corpus of argumentative microtexts. In: ECA, vol 2. College Publications, Lisbon, pp 801–816
24. Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: EMNLP, pp 1532–1543 https://doi.org/10.3115/v1/D14-1162
25. Richter MM, Weber RO (2013) Case-based reasoning – A textbook. Springer, Heidelberg, Berlin, New York https://doi.org/10.1007/978-3-642-40167-1
26. Robertson SE, Zaragoza H (2009) The probabilistic relevance framework: BM25 and beyond. Found Trends Inf Retr 3(4):333–389. https://doi.org/10.1561/1500000019
27. Salton G, Wong A, Yang C (1975) A vector space model for automatic indexing. Commun ACM 18(11):613–620. https://doi.org/10.1145/361219.361220
28. Speer R, Chin J, Havasi C (2017) Conceptnet 5.5: An open multilingual graph of general knowledge. In: AAAI, pp. 4444–4451. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972. Accessed 06th June 2020
29. Wachsmuth H, Potthast M, Khatib KA, Ajjour Y, Puschmann J, Qu J, Dorsch J, Morari V, Bevendorff J, Stein B (2017) Building an argument search engine for the web. In: Workshop argument mining, pp 49–59 https://doi.org/10.18653/v1/W17-5106
30. Walton D (2006) Fundamentals of critical argumentation. Cambridge University Press, Cambridge https://doi.org/10.1017/CBO9780511807039
31. Wu Z, Palmer M (1994) Verbs semantics and lexical selection. Proc. of the 32nd Ann. Meeting on Ass. for Comp. Linguistics.