

Similarity Measures for Case-Based Retrieval of Natural Language Argument Graphs in Argumentation Machines

Ralph Bergmann and Mirko Lenz and Stefan Ollinger and Maximilian Pfister

Business Information Systems II

University of Trier

54296 Trier, Germany

www.wi2.uni-trier.de

bergmann@uni-trier.de, info@mirko-lenz.de, ollinger@uni-trier.de

Abstract

In the field of argumentation, the vision of robust argumentation machines is investigated. They explore natural language arguments 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. We aim at combining methods from case-based reasoning (CBR), information retrieval, and computational argumentation to contribute to the foundations of argumentation machines. In this paper, we focus on the retrieval phase of a CBR approach for an argumentation machine and propose similarity measures for arguments represented as argument graphs. We evaluate the similarity measures on a corpus of annotated micro texts and demonstrate the benefit of semantic similarity measures and the relevance of structural aspects.

1 Introduction

Argumentation is a core activity in human communication in all fields of life. Researchers, journalists, or human decision makers often aim at obtaining a comprehensive overview of current arguments and opinions related to a certain topic. They intend to develop personal, well-founded opinions justified by convincing arguments. Therefore, traditional search engines are largely used to gather textual information, which is then manually analyzed to extract, understand, evaluate, and summarize the arguments contained therein. To overcome the limited support by search engines for this task, the vision of *argumentation machines* (Reed and Norman 2003) emerged, which is also the topic of the special research program RATIO¹ funded by the German Research Foundation. Such argumentation machines automatically explore and process available information sources on the Web, particularly argumentative texts and factual content relevant for the specific topic under discussion. Unlike existing search engines, which operate on the textual level, such argumentation machines will reason on a knowledge level formed by arguments. For a given particular context, such reasoning will support the deliberation of arguments and counter-arguments for the issue under consideration and in addition it could support the synthesis of new arguments, based on analogical transfer from similar ones.

In our work, carried out as part of the ReCAP project (Bergmann et al. 2018), we aim at combining methods from case-based reasoning (CBR), information retrieval (IR), and computational argumentation (CA) to contribute to the foundations of argumentation machines. CA provides the foundation for representing arguments on the knowledge level. A (whole) argument consists of single sub-arguments substantiating the plausibility of the main argument making use of rhetorical devices (Walton, Reed, and Macagno 2008) and can be formally represented as a graph (Bex, Prakken, and Reed 2010). CBR (Aamodt and Plaza 1994; Brantken 2003; Weber, Ashley, and Brüninghaus 2005) can support the reasoning with arguments by similarity-based retrieval of arguments relevant for a user's topic and the subsequent adaptation to support the synthesis of new arguments. In addition IR supports the selection of relevant documents (Stab et al. 2018) and the validation of the factual correctness of individual claims (Leong and Cucerzan 2012). Both CBR and IR have a long tradition in the development of similarity measures for retrieval, which is an excellent starting point for the development of similarity measures for arguments.

In this paper, we address the problem of developing similarity measures for arguments represented as argument graphs for the purpose of argument retrieval. Thus, we focus on the retrieval phase of a CBR approach for an argumentation machine that reasons with arguments stored as cases in a case base. In particular, we focus on argument retrieval with structured queries in form of argument graphs. A query can also represent a partial argument leaving out parts, e. g. its conclusion in order to transfer it from the most similar argument. The proposed similarity measure is derived from previous work on process-oriented CBR (POCBR), in which the similarity of graphs is assessed that represent semantically annotated workflows (Bergmann and Gil 2014). The similarity of individual natural language propositions of the argument graph is based on word embeddings. Further, a MAC/FAC retrieval approach (Forbus, Gentner, and Law 1995) is developed to improve accuracy and computational performance of retrieval. We evaluate the method on a corpus of annotated micro texts containing different topics.

Next, we present related work in the field. Section 3 introduces the similarity measures and the retrieval approach. Various variants are evaluated in Section 4. Finally, Section 5 concludes the paper.

2 Foundations and Related Work

While there is a significant amount of work in CA on the theoretical properties of argumentation logics, the development of practical systems for real-life argumentation requires a representation of arguments which is expressive enough to capture the complexity of human argumentation. Such representations define arguments and their interaction. Arguments are structured in the sense that they provide a set of premises and a claim that follows from the premises (Walton, Reed, and Macagno 2008). Arguments can be attacked by other arguments, often in the form of rebuttals (attacking the claim) or undercuts (attacking the premises of the argument and how they support the claim). In this sense, the Argument Interchange Format (AIF) (Chesnevar et al. 2006; Bex, Prakken, and Reed 2010) allows the representation of natural-language arguments in the form of an argument graph consisting of certain kind of nodes.

Figure 1 shows an example of an argumentation in the AIF format. It describes a point of view to a political topic in a city and opposing and supporting opinions.

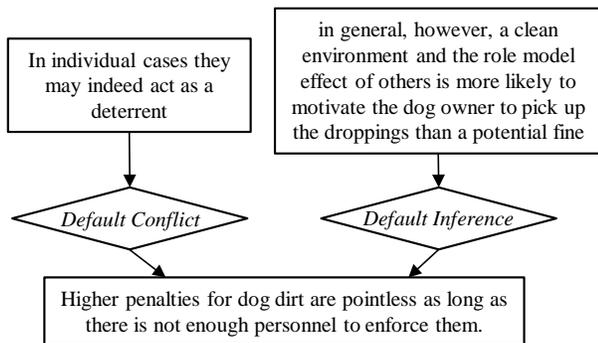


Figure 1: Example of an argument graph in AIF format from the Microtexts corpus (Peldszus and Stede 2015)

While argument mining methods (Lippi and Torroni 2016) aim at converting natural language argumentative texts into such argumentation graphs, our work aims at supporting the reasoning with such graphs. Due to the limitations of formal argumentation frameworks (Caminada and Wu 2011) we believe that such logic-based reasoning approaches are of limited use for future argumentation machines reasoning with real-world arguments. Thus, we propose CBR as it does not require a complete and consistent domain theory and is able to make use of vague information. In particular, it is not based on a crisp notion of truth, but on notions of similarity and utility (Richter and Weber 2013). In particular, textual CBR (Weber, Ashley, and Brüninghaus 2005) is highly relevant, as it deals with case representations based on natural language texts. It is being used since the 1980s in the context of argumentation for legal reasoning (Ashley 1988; Branting 2003). Research on CBR for legal argumentation is based on a model of legal argument, which has recently also been formalized in terms of the ASPIC+ argumentation framework (Prakken et al. 2015). Cases are represented based on hierarchically

structured factors or issues (Rissland, Ashley, and Branting 2005), which are used during similarity-based retrieval. A factor is similar to an argument or premise, having an abstract label corresponding to an argument scheme (Walton, Reed, and Macagno 2008). The similarity of two arguments is defined by the commonalities and differences of the factors. CATO (Alevi 1997) extends those argument graphs with intermediate factors, forming a factor hierarchy. Further, case justifications are reused and adapted to create new arguments in a case-based manner (Branting 2003). In CA, similar ideas have been recently established such as the “recycling” of arguments for synthesis of claims (Bilu and Slonim 2016), which is essentially the idea behind CBR.

Searching arguments is also addressed in the field of IR. For argument retrieval most works support textual queries. Recent work allows to retrieve a set of arguments relevant to a topic from a large corpus (Stab et al. 2018). The indexed documents are scored against a keyword query. Argument identification and stance recognition filter claims from the resulting documents. Arguments consisting of a set of claims for a topic can be found in a corpus of text documents (Gutfreund, Katz, and Slonim 2016). A semantic analysis of the query is done, e. g. extraction of key concepts, stance recognition towards the key concepts and query expansion. Other work propose the use of an index of argument structures (Wachsmuth et al. 2017) to enable efficient querying.

3 Case-Based Retrieval of Argument Graphs

We now describe our approach for similarity-based argument graph retrieval. Our main motivation is to achieve a ranking of arguments that is in line with a human expert. Thus, we aim at developing a similarity measure that considers the semantics of the textual descriptions of the nodes as well as the overall structure of the argument graph.

Representation of Argument Graphs

We developed a case representation using argument graphs, which is inspired by the graph representation of AIF as well as a graph representation used in POCBR (Bergmann and Gil 2014). It is similar to text reasoning graphs as proposed by Sizov et al. (2014) for representing causal information, but they contain in addition semantic information in different forms. An argument graph is a 5-tuple $W = (N, E, \tau, \lambda, t)$ in which N is a set of nodes, $E \subseteq N \times N$ is a set of edges, $\tau: N \rightarrow \mathcal{T}$ is a function that maps nodes to types, and $\lambda: N \rightarrow \mathcal{L}$ is a function that maps nodes to labels representing the textual content. In addition $t \in \mathcal{L}$ is a label, describing the overall topic of the argument represented in the graph. The node types \mathcal{T} are specified according to the type ontology used in AIF. We distinguish on the top level between Information (I) nodes and Scheme (S) nodes. I-nodes contain textual information (see Figure 1) whereas S-nodes are used to characterize the relation between two I-nodes. S-nodes are further split into sub-types, including Rule of Inference Application (RA) nodes and Conflict Application (CA) nodes. The node types Default Conflict and Default Inference shown in Figure 1 are further specializations of CA and RA nodes, respectively.

The mapping function λ is used to link a representation of the node text to each I-node. We chose a vector representation for \mathcal{L} in order to encode a semantic representation of the node text, i.e., a pre-computed embedding vector. Also t is such a vector representing the semantics of the whole graph.

For retrieval, a case base of argument graphs (resulting from an existing corpus or created by argument mining) is constructed. We also consider a query to be an argument graph or a fraction of it. In particular, a query can also consist of a single I-node, leading to the type of queries traditionally considered by IR methods for argument retrieval.

Semantic Representation and Similarity of I-Nodes

A main issue to be considered when creating the argument graphs to be stored in the case base is the construction of the labels for the I-nodes as well as the label t . For the I-nodes, the node text is pre-processed by tokenization and stopword elimination. In addition, the IDF score (Salton and Buckley 1988) for each token is calculated by $idf_{tk,D} = \log_{10}(|D|/f_{tk,D})$, with $f_{tk,D}$ representing the number of times token tk appears in the case base. To construct the vector representation of the text, we use the semantic word embedding method word2vec (Mikolov et al. 2013) to transform each token into an embedding vector. These vectors are then averaged to construct the overall vector $\lambda(n)$ for each I-node n . Here, the IDF score can be used to apply a weighted average. The same approach, applied to the concatenation of all I-node texts in the graph, is used to construct the vector t representing the overall topic of the argument graph.

The semantic representation is then used to determine the semantic similarity of different I-nodes by applying the traditional cosine similarity measure. With the cosine measure applied to the topic vectors t it is also possible to compute a kind of semantic similarity between two whole argument graphs. This similarity is however, purely based on the textual content, totally neglecting the structure of the argument.

In order to later evaluate to benefit of the semantic similarity, a lexical similarity measure for I-nodes and the graph topic t can be used as well, which does not make use of the vector representation, but which operates directly on the text using a traditional Levenshtein distance.

Graph-based Argument Similarity

In order to determine the similarity between two argument graphs in a way that the structure of the argumentation is considered, we transfer the approach for a graph-based semantic workflow similarity proposed by Bergmann & Gil (2014). It is based on the local-global-principle (Richter and Weber 2013), determining the local similarities for the nodes and edges in the graphs to be compared which are then combined to a global similarity. The similarity $sim(QA, CA)$ between a query argument QA and a case argument CA is defined by means of an admissible mapping $m : N_q \cup E_q \rightarrow N_c \cup E_c$, which is a type-preserving, partial, injective mapping function of the nodes and edges of QA to those of CA . An edge can only be mapped if the nodes that the edge connects are also mapped to the respective nodes which are linked by the mapped edge. For each query node and

edge x mapped by m , the similarity to the respective case node or edge $m(x)$ is computed by $sim_N(x, m(x))$ and $sim_E(x, m(x))$, respectively. For comparing two I-nodes, sim_N is the semantic similarity measure described above, while the similarity of S-nodes is determined as an exact match of the node type, i.e., the similarity is 1 if the node is of the same type and 0 otherwise. Comparing an I-node with an S-node also leads to a similarity of 0. Further, sim_E is the average of the sim_N values of the two nodes linked by the edge. The similarity with respect to a mapping m , named $sim_m(QA, CA)$ is computed by a weighted average combining the similarity values of all mapped nodes and edges. Finally, the overall argument similarity is determined by the best possible mapping m

$$sim(QA, CA) = \max\{sim_m(QA, CA) \mid \text{admis. map } m\}.$$

This similarity measure assesses how well the query argument is covered by the case argument. In particular, the similarity is 1 if the query argument is exactly included in the case argument as a sub-graph.

The computation of the argument similarity requires solving the involved optimization problem, which is done using the A*-algorithm named A*I proposed by Bergmann & Gil (2014). It is important to note that the result of this similarity assessment is not only the similarity value, but also the resulting mapping, which represents a connection across two argument graphs that can form the foundation for argument adaptation as part of a complete CBR approach.

Retrieval of Argument Graphs

One major drawback of the A*I mapping algorithm is its complexity. This is a problem when dealing with large graphs and large case bases. A linear retrieval approach that sequentially computes the similarity w. r. t. each case in the case base can lead to unacceptable retrieval times. In process-oriented CBR, this issue has already been successfully addressed (Bergmann and Stromer 2013) by a MAC/FAC (*many are called, but few are chosen*) approach (Forbus, Gentner, and Law 1995). More specifically, this means dividing the retrieval into a very efficient pre-filter stage (MAC phase) and the subsequent FAC phase, in which only the filtered cases are assessed using the complex similarity measure.

We propose a MAC/FAC approach for argument graphs in which the MAC phase is implemented as a linear similarity-based retrieval of the cases based on the semantic similarity of topic vector t only. The filter selects the k most similar cases, which are passed over to the FAC phase which implements the ranking by a linear assessment of the cases using the graph-based similarity as described above. We expect that this MAC/FAC approach can reduce the retrieval time if the MAC phase shows a good recall for a relatively small value of k . We also hope that if it succeeds to produce a high precision that it can discard cases from argument domains that are totally different from the current query. This could also help to avoid problems with the semantic matching of the I-nodes that could arise due to the fact that the size of the text in the I-nodes is relatively short and thus relevant context information might not be available.

4 Experimental Evaluation

The described retrieval approach has been implemented in Python 3. The developed system comes with a web interface that allows to specify a query and to configure certain parameters of the similarity measures. For the following evaluation, we evaluate three variants of retrieval a) the linear MAC retrieval using only the similarity of the topic vector t , b) the graph-based linear retrieval over the full case base, as well as c) the MAC/FAC retrieval.

Hypotheses

We performed a systematic evaluation to test how well the three proposed approaches are able to retrieve and rank cases in a way that in line with the assessment of a human expert. The following three hypotheses are subject of this evaluation.

- Hypothesis H1: The graph-based retrieval and MAC/FAC retrieval lead to a better alignment with an expert’s assessment than the MAC retrieval only.
- Hypothesis H2: The semantic similarity measure based on word embeddings leads to results that are better aligned with the expert’s assessment than a standard lexical similarity measure.
- Hypothesis H3: The MAC/FAC approach increases the accuracy and the computational performance of the retrieval compared to the pure graph-based approach.

Experimental Setup

We used the corpus of annotated microtexts published by Peldszus and Stede (2015)². It consists of argument graphs of high quality about diverse topics, which were created manually by experts. The corpus and thus our case base, consists of 110 argument graphs with a total of 576 I-nodes, 272 RA-nodes, and 171 CA-nodes.

For the purpose of this evaluation, several experts with knowledge in the domain of the corpus created a set of queries using the OVA tool³. Each expert was asked to create four queries for a certain topic covered by the corpus. The topics have been selected such that the corpus contains about six to eight argument graphs relevant for the topic. In total six topics have been identified, leading to a total of 24 queries. The number of I-nodes in the queries varies between one and three. The reference cases for each query (our gold standard) were also selected and ranked by the same experts. The ranking determined by the experts explicitly allowed multiple cases to have the same rank.

To compute the semantic similarity with word embeddings, a pre-trained word2vec model trained on Google News (Mikolov et al. 2013) is used⁴.

We performed several retrieval experiments with the different algorithms and determine the resulting values for Precision (P), Recall (R), and F-Score (F_{β}).

²<https://github.com/peldszus/arg-microtexts>

³<http://ova.arg-tech.org>

⁴<https://code.google.com/archive/p/word2vec/>

Additionally, the following ranked metrics Average Precision (AP) and Correctness (Cor) were computed, defined as follows:

$$AP = \frac{1}{|\{\text{relevant}\}|} \cdot \sum_{k=1}^n P_k \cdot \text{rel}(k) \quad \text{Cor} = \frac{c - d}{c + d}$$

The function $\text{rel}(k)$ is 1 if case k is relevant and 0 otherwise; P_k is the precision at cut-off k in the list of the retrieved n cases.

The second equation compares the ranking order of the retrieval with a reference order which is created by human experts. Based on these orders, the concordance c is the number of items that are ordered equally. The discordance d on the other hand measures the number of items that are ordered differently (Cheng et al. 2010).

Results and Discussion

We now present the results of the evaluation. In the following tables, the metrics are shown in columns while different runs of the system are shown in rows. It is important to note that all evaluation metrics are averaged over all queries using the arithmetic mean. Table 1 shows the results of the MAC

Table 1: Results of unranked evaluation of the MAC retrieval using different filter sizes k

Size k	P	R	F_1	F_2
5	0.933	0.661	0.771	0.701
10	0.654	0.912	0.759	0.842
15	0.464	0.971	0.626	0.794
20	0.354	0.988	0.52	0.725

retrieval using different filter sizes k . As the MAC retrieval is primarily used as a pre-selection approach, we show the results of the unranked metrics only. It is evident that the precision is very high for a small k and that recall is increasing towards 1 quite fast. It is important to note that for each query only six to eight cases are relevant in total, thus increasing k to higher values inevitably leads to a decrease in precision. For the given case base, a value of $k = 10$ is the best choice, leading to high recall and acceptable precision. Also the F -measures have their highest value for this value of k . Consequently, we will use $k = 10$ for all subsequent experiments.

Table 2 shows the results of the three retrieval methods each of which is used with the lexical and the semantic similarity approach for comparing textual information. Clearly,

Table 2: Comparison of the three retrieval algorithms with semantic and lexical similarity

Algorithm	Sim	AP	Cor
MAC	lex.	0.084	-0.125
MAC	sem.	0.884	0.544
Graph	lex.	0.337	0.268
Graph	sem.	0.768	0.539
MAC/FAC	lex.	0.1	0.042
MAC/FAC	sem.	0.857	0.581

the semantic similarity measure outperforms the lexical similarity measure in all algorithms. This is an expected result due to polysemy and a distributed representation which means that hypothesis H2 can be accepted. The lexical similarity performs best in the graph-based approach, which is most likely caused by the comparison of relatively short texts in the I-nodes instead of the text of the entire graph. The latter leads to a very high edit distance and thus the system is unable to create a correct ranking. Surprisingly, the semantic similarity when used within the graph-based algorithm, does not lead to average precision and correctness results which exceed the MAC approach. One reason might be that an embedding of the full text (i. e. the topic vector) can capture more contextual information than the embeddings of the short texts in the I-nodes. However, MAC/FAC produces the best results for the semantic similarity according to the correctness measure. This is possibly due to the fact that the MAC phase ensures that the graph-based retrieval only works on thematically filtered case, thereby avoiding inappropriate mappings otherwise caused by the weaker semantic similarity assessment of the short texts in the I-nodes. In summary, we can see that Hypothesis H1 could only be partially accepted. For the MAC/FAC algorithm we can see that Hypothesis H3 is confirmed with respect to the higher accuracy compared to the graph-based approach.

Table 3 evaluates queries of varying size. For 1-node and 2-node queries, the average precision values are quite in the same range for all three algorithms, however the MAC/FAC approach outperforms the pure MAC and the graph-based approach concerning the correctness of the ranking. For 3-node queries it is quite surprising that the text-based retrieval approach is clearly the best, although the query has a stronger graph structure. The reason for this is not yet fully clear, but it might be due to the fact that only a very small number of 3-node queries are available, thus this particular result is not significant.

Table 3: Results of evaluation using queries of different sizes

Algorithm	Query	AP	Cor
MAC	1-node	0.854	0.599
Graph	1-node	0.849	0.635
MAC/FAC	1-node	0.851	0.684
MAC	2-node	0.916	0.515
Graph	2-node	0.899	0.450
MAC/FAC	2-node	0.913	0.567
MAC	3-node	0.908	0.495
Graph	3-node	0.437	0.277
MAC/FAC	3-node	0.799	0.363

Finally, we evaluated whether traditional pre-processing methods from IR can help to further improve the results. Table 4 shows the impact of the IDF score as well as the exclusion of stopwords on the ranking produced by the MAC/FAC approach. Two different versions of IDF have been evaluated: graph and node. The graph-based method treats the full text of each case as one document, while the node-based approach uses the texts in each I-node separately. Further, stopwords can either be excluded or left in the text (included).

Table 4: Results of evaluation of the MAC/FAC algorithm using different preprocessing options

IDF	Stopwords	AP	Cor
none	include	0.857	0.581
none	exclude	0.91	0.526
graph	include	0.859	0.512
graph	exclude	0.885	0.481
node	include	0.853	0.506
node	exclude	0.856	0.459

The results indicate that the exclusion of stopwords has a positive effect on the average precision. However, the correctness is highest when not using any pre-processing option at all. This also justifies the previous experiments reported, which all have been performed without pre-processing. The runtime heavily depends on the method used. On a 2014 MacBook Pro 15" with a 2.8 GHz Intel Core i7 processor and 16 GB of RAM the pure MAC retrieval can be performed within a second. The linear graph-based retrieval takes up to 20 minutes, thus the expected complexity problem is clearly visible and a justification for the introduction of the MAC/FAC approach. Using MAC/FAC, the processing time comes down to under one minute. Together with the results from table 2 and 3, Hypothesis H3 can be clearly accepted.

5 Conclusion

In this paper, we proposed a graph-based similarity measure for the case-based retrieval of argument graphs which is used as part of a MAC/FAC retrieval method. The graph-based approach is able to consider the structure of the argument graph when assessing the similarity to the query graph. It results in a similarity assessment together with a mapping of the query nodes to nodes in the retrieved case, which we expect to be important information for a future adaptation of the argument graph in order to enable argument synthesis. Our results demonstrate the advantages of the MAC/FAC retrieval approach in terms of retrieval quality and computational performance. However, it was surprising that the pure text-based retrieval outperforms the graph-based approach in 3-node queries. Future work will involve more comprehensive experimental evaluations, for which we currently invest a significant amount of work into the development of new argument corpora and related use cases from the political domain. We will also extend our experimental work in order to evaluate additional approaches for semantic similarity of I-nodes by using advanced embedding methods. Also the similarity assessment of S-nodes can be improved by including a more fine-grained comparison of different S-node sub-types. On the more challenging side, new methods for the adaptation of argument graphs need to be developed which make use of the mappings found during similarity assessment. Therefore, we aim at transferring compositional adaptation methods from POCBR (Müller and Bergmann 2014) as they are also based on a graph-based case representation.

Acknowledgments

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

References

- Aamodt, A., and Plaza, E. 1994. Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Communications* 7(1):39–59.
- Aleven, V. 1997. *Teaching Case-Based Argumentation Through a Model and Examples*. Ph.D. Dissertation, University of Pittsburgh.
- Ashley, K. D. 1988. *Modelling Legal Argument: Reasoning with Cases and Hypotheticals*. Ph.D. Dissertation, University of Massachusetts, Amherst, MA, USA. Order No: GAX88-13198.
- Bergmann, R., and Gil, Y. 2014. Similarity assessment and efficient retrieval of semantic workflows. *Information Systems* 40:115–127.
- Bergmann, R., and Stromer, A. 2013. MAC/FAC retrieval of semantic workflows. In *Proc. of the 26th FLAIRS Conference*. AAAI Press.
- Bergmann, R.; Schenkel, R.; Dumani, L.; and Ollinger, S. 2018. 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, Mannheim, Germany, August 22-24, 2018.*, volume 2191 of *CEUR Workshop Proceedings*. CEUR-WS.org.
- Bex, F.; Prakken, H.; and Reed, C. 2010. A formal analysis of the AIF in terms of the ASPIC framework. In *Computational Models of Argument: Proceedings of COMMA 2010, Desenzano del Garda, Italy, September 8-10, 2010*, 99–110. IOS Press.
- Bilu, Y., and Slonim, N. 2016. Claim synthesis via predicate recycling. In *Proc. 54th Annual Meeting of the Association for Computational Linguistics (ACL)*. The Association for Computer Linguistics.
- Branting, K. 2003. A reduction-graph model of precedent in legal analysis. *Artificial Intelligence* 150(1):59–95.
- Caminada, M., and Wu, Y. 2011. On the limitations of abstract argumentation. In *Proceedings of the 23rd Benelux Conference on Artificial Intelligence*. Neveland, Drogenen.
- Cheng, W.; Rademaker, M.; De Baets, B.; and Hüllermeier, E. 2010. Predicting partial orders: ranking with abstention. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 215–230. Springer.
- Chesnevar, C.; Modgil, S.; Rahwan, I.; Reed, C.; Simari, G.; South, M.; Vreeswijk, G.; Willmott, S.; et al. 2006. Towards an argument interchange format. *The knowledge engineering review* 21(4):293–316.
- Forbus, K. D.; Gentner, D.; and Law, K. 1995. MAC/FAC - A model of similarity-based retrieval. *Cognitive Science* 19(2):141–205.
- Gutfreund, D.; Katz, Y.; and Slonim, N. 2016. Automatic arguments construction – from search engine to research engine. In *2016 AAAI Fall Symposium Series*. AAAI Press.
- Leong, C. W., and Cucerzan, S. 2012. Supporting factual statements with evidence from the web. In *Proc. 21st ACM Int. Conf. on Information and Knowledge Management (CIKM)*, 1153–1162.
- Lippi, M., and Torrioni, P. 2016. Argument mining from speech: Detecting claims in political debates. In *Proc. 13th AAAI Conf on Artificial Intelligence*, 2979–2985. AAAI Press.
- Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv:1301.3781 [cs]*.
- Müller, G., and Bergmann, R. 2014. Workflow streams: A means for compositional adaptation in process-oriented CBR. In *Proc. of the 22th Int. Conf. on Case-Based Reasoning*, volume 8766 of *LNCS*, 315–329. Springer.
- Peldszus, A., and Stede, M. 2015. An annotated corpus of argumentative microtexts. In *First European Conference on Argumentation: Argumentation and Reasoned Action, Portugal, Lisbon*.
- Prakken, H.; Wyner, A. Z.; Bench-Capon, T. J. M.; and Atkinson, K. 2015. A formalization of argumentation schemes for legal case-based reasoning in ASPIC+. *J. Log. Comput.* 25(5):1141–1166.
- Reed, C., and Norman, T. 2003. *Argumentation machines: New frontiers in argument and computation*. Springer.
- Richter, M. M., and Weber, R. O. 2013. *Case-Based Reasoning - A Textbook*. Springer.
- Rissland, E. L.; Ashley, K. D.; and Branting, K. 2005. Case-based reasoning and law. *The Knowledge Engineering Review* 20(3):293–298.
- Salton, G., and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 24(5):513–523.
- Sizov, G.; Öztürk, P.; and Štyrák, J. 2014. Acquisition and reuse of reasoning knowledge from textual cases for automated analysis. In Lamontagne, L., and Plaza, E., eds., *Case-Based Reasoning Research and Development*, 465–479. Springer.
- Stab, C.; Daxenberger, J.; Stahlhut, C.; Miller, T.; Schiller, B.; Tauchmann, C.; Eger, S.; and Gurevych, I. 2018. Argumenttext: Searching for arguments in heterogeneous sources. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 21–25. Association for Computational Linguistics.
- Wachsmuth, H.; Potthast, M.; Al Khatib, K.; Ajjour, Y.; Puschmann, J.; Qu, J.; Dorsch, J.; Morari, V.; Bevendorff, J.; and Stein, B. 2017. Building an argument search engine for the web. In *Proceedings of the 4th Workshop on Argument Mining*, 49–59. The Association for Computer Linguistics.
- Walton, D.; Reed, C.; and Macagno, F. 2008. *Argumentation Schemes*. Cambridge University Press.
- Weber, R. O.; Ashley, K. D.; and Brüninghaus, S. 2005. Textual case-based reasoning. *The Knowledge Engineering Review* 20(03):255–260.