



The Road Map to FAME: A Framework for Mining and Formal Evaluation of Arguments

Ringo Baumann¹ · Gregor Wiedemann² · Maximilian Heinrich² · Ahmad Dawar Hakimi² · Gerhard Heyer²

Received: 5 February 2020 / Accepted: 30 May 2020 / Published online: 16 June 2020
© The Author(s) 2020

Abstract

Two different perspectives on argumentation have been pursued in computer science research, namely approaches of argument mining in natural language processing on the one hand, and formal argument evaluation on the other hand. So far these research areas are largely independent and unrelated. This article introduces the agenda of our recently started project “FAME – A framework for argument mining and evaluation”. The main project idea is to link the two perspectives on argumentation and their respective research agendas by employing controlled natural language as a convenient form of intermediate knowledge representation. Our goal is to develop a framework which integrates argument mining and formal argument evaluation to study patterns of empirical argumentation usage. If successful, this combination will allow for new types of queries to be answered by argumentation retrieval systems and large-scale content analysis. Moreover, feeding evaluation results as additional knowledge input to argument mining processes could be utilized to further improve their results.

Keywords Argument Mining · Argument Evaluation · Controlled Natural Languages

1 Two Worlds of Argumentation

The study of argumentation has attracted increasing attention by computer scientists over the past decade. Two major strands can be distinguished. On the one hand, argument mining as a sub-field of natural language processing (NLP) strives to identify, classify and relate argumentative structures in texts. On the other hand, researchers from theoretical computer science investigate how to model argumentation with symbolic knowledge representations that can be logically evaluated with a formalism. While the former profit from the availability of large amounts of empirical data thanks to the digitization of communication in the internet era, they struggle to incorporate more complex, symbolic knowledge into their works. The latter, while having developed expressive models and tools for logical represen-

tation and reasoning of formal argumentation, largely lack empirical grounding or application of their works. In this light, a large mutual benefit could be expected from a combination of both research strands.¹ However, until recently the two research communities remained surprisingly disconnected, mutually neglecting their publications and rarely meeting for joint events. With our research project “FAME – A framework for argument mining and evaluation”, we strive to bridge the gap between those two worlds. Employing controlled natural languages (CNLs) as an intermediate representation for argumentation, we will investigate ways to transform the empirical use of arguments into a machine-evaluable form and explore the potentials of automatic logical reasoning for the analysis of argumentation at large scale.

This article introduces the fundamental ideas, selected technologies and targeted goals of our recently started project.² For this, the upcoming section reflects on what we consider as an argument against the background of

Ringo Baumann
baumann@informatik.uni-leipzig.de

✉ Gregor Wiedemann
gwiedemann@informatik.uni-hamburg.de

¹ Leipzig University, Augustusplatz 10, 04109 Leipzig, Germany

² Hamburg University, Vogt-Kölln-Str. 30, 22527 Hamburg, Germany

¹ This is also a major hypothesis underlying the Priority Programme “Robust Argumentation Machines” (RATIO, SPP 1999) funded by the German Research Foundation (DFG) since 2017.

² The project started in summer 2019 and, due to the thematic fit, was given the opportunity to join the DFG Priority Programme RATIO, SPP 1999.

the two research fields involved. The third section introduces formal argument evaluation. In the fourth section, our approach to using CNL as intermediate representation is described along with illustrative examples. Finally, we explain the planned architecture and give an outlook on the expected outcomes of our project.

2 Argumentation in NLP

For a couple of years now, the study of argumentation in natural language attracts lots of attention from researchers. In a seminal paper by Palau and Moens [14], argument mining is introduced as a series of consecutive NLP tasks for automatic extraction of argument structures from unstructured documents such as newspaper texts, blogs, or user comments. The chain of tasks comprises 1) argument unit segmentation of the input text, 2) classification of units into functional types such as premise or claim, and 3) the classification of structures between them, e.g. whether they support or attack each other or are pro/contra regarding a certain topic. Single steps usually rely on supervised machine learning in which a classification model is trained based on a manually annotated corpus. Corpora are annotated with different annotation schemes of varying complexity. Simpler approaches are claim detection [13] or stance classification [1] where pro/con positions towards certain issues are classified. More complex approaches try to adapt theoretically derived annotation schemes such as the Toulmin model [16] or Walton schemes [20]. In these studies, the complexity of annotation schemes usually is reduced by synthesizing subsets of the theoretically defined categories [15].

During the conceptualization of our project, we identified two major gaps in the current research on NLP-based argument mining. *First*, the field lacks a common definition of its subject and suffers from simplification of its operationalization. Although most works acknowledge a semantic core of an argument, i.e. that a premise is required to underpin the plausibility of a claim [17], supervised learning to detect such functional types of argument units usually boils down to statistical learning of superficial language cues. This is because high quality of training data for machine learning requires high inter-annotator agreement for the argument constituents. Usually, this can only be achieved by narrowing down either genre (e.g. student essays), topic (e.g. gay marriage), or argument type (e.g. study evidence) for the manual annotation task. Still, annotation is costly such that typical argument mining corpora comprise only some hundred to a few thousand texts. Any model simply striving for the detection of functional types based on manually labeled resources will eventually overemphasize corpus-specific language patterns such as topic words or

discourse markers [8], and neglect semantic features as well as their logical dependencies. Consequently, generalization of trained models to new datasets is poor, as can be tested with tool Targer by Chernodub et al. [6]. *Second*, since argument mining often boils down to automatic classification of functional argument unit types based on linguistic patterns, it refrains from the incorporation of any semantic knowledge. Unfortunately, this impedes not only the use of structured background knowledge, semantic priors or logical constraints which could provide helpful information to the mining process. It also prevents interesting use cases of automated evaluation of empirical argumentation.

In light of these shortcomings, we argue that a practically useful argument mining system requires a more expressive encoding of arguments based on semantics. This would also allow modeling implicit background assumptions which often cannot be observed from empirical arguments directly, but can be encoded in knowledge bases (KB) curated by domain experts. In our project, we explore ways and opportunities for using such a KB for argument mining and formal evaluation.

3 Formal Argument Evaluation

Computational models of *argumentation* have become one central topic in leading AI conferences. Moreover, since 2006, there is a biennial conference called *Computational Models of Argument*. The formal analysis of argumentation studies how to model arguments and their relationships, as well as the necessary conflict resolution in the presence of diverging opinions. One can distinguish two major lines of research in the field: logic-based and abstract approaches. The former takes the logical structure of arguments into account and defines notions like attack, undercut, defensibility, etc. in terms of logical properties of the chosen argument structures (cf. [3, 4] for excellent overviews). For instance, in case of propositional logic, an argument is a premise/claim pair $A = (\Phi, \phi)$ where Φ is a set of formulas and ϕ a single propositional formula, s.t. (i) $\Phi \models \phi$, (ii) Φ is consistent and (iii) Φ is \subseteq -minimal w.r.t. (i) and (ii). The pairs $(\{p, \neg q \rightarrow \neg p\}, q)$ and $(\{(s \vee q) \rightarrow t, \neg t\}, \neg s \wedge \neg q)$ are examples of logical arguments. Moreover, we may say that both arguments attack each other since the union of their claims, namely $\{q, \neg s \wedge \neg q\}$ is inconsistent.

Abstract approaches, in contrast, abstract away from the internal structure of arguments and consider them as atomic items, focusing entirely on the attack relation among arguments. This means it is assumed that the reason why something is an argument was already identified beforehand. Such a reason can be an explicit construction from a given background KB as sketched above or simply an argument mining process applied to real-world data. This

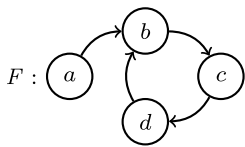


Fig. 1 Example of a simple argumentation framework

means the abstract approach is not “standalone”, it rather depends on methods for generating abstract arguments in the first place and a subsequent instantiation process. The main aim of abstract argumentation is to provide possibilities to evaluate conflicting scenarios, i.e. to return reasonable sets of arguments that represent *acceptable* positions one may take in the light of the available arguments.

From AFs to ADFs: At the heart of the abstract approach are currently Dung’s widely used *argumentation frameworks* (AFs) [7] and their associated semantics. In a nutshell, an AF is a directed graph $F = (A, R)$ with a set of vertices A being the abstract arguments and a set of directed edges $R \subseteq A \times A$ corresponding to attacks between arguments. Consider the AF F in Fig. 1 consisting of four arguments a, b, c , and d .

Such a conflicting scenario is resolved by so-called *semantics*. By now a variety of argumentation semantics has been defined, each one encoding different desiderata for acceptable sets of arguments, so-called *extensions* (cf. [2, Sect. 3.1.2] for a compact summary). One of the most prominent ones was already defined by Dung in 1995, so-called *stable semantics*. Informally, a set of arguments is a stable extension if there are no conflicts between them and all other arguments are attacked by at least one argument of the set. This means, in case of F we obtain one single stable extension, namely $E = \{a, c\}$. The position E does not contain any conflict and, furthermore, all remaining arguments (i.e. b and d) are refuted. Such positions are very desirable in a debate or argumentation scenarios in general.

Dung’s argumentation frameworks, as well as most of the available semantics, are very intuitive and easy to understand. However, they suffer from various shortcomings. One main drawback is that they are rather limited in their expressive capabilities implying that they are not necessarily the right target systems for instantiation. More precisely, modeling the relations between arguments is problematic if these relations are more complex than a simple binary attack between two arguments. For this reason, *Abstract Dialectical Frameworks* (ADFs) were introduced [5]. They are a natural generalization of classical Dung-style AFs. The fundamental idea behind them is to stick to abstract arguments which remain atomic entities and thus are not further analyzed as in AFs, yet to allow for much more flexible relationships among arguments. In particular, arguments can not only attack each other, but they also may pro-

vide support for other arguments which is not expressible in classical AFs. This expressive power is achieved by adding acceptance conditions to the arguments which allow for the specification of arbitrary relationships between arguments and their parents in the argument graph. Acceptance conditions are usually expressed in terms of propositional formulas. Consequently, an ADF can be represented as a pair $D = (S, \Phi)$ where S is a set of *statements* and Φ a set of *acceptance functions*. A semantics-preserving translation from AFs to ADFs takes former arguments as statements and formalizes the acceptance function of a certain statement a as the conjunction of all negated attackers of a . This means a statement a can be accepted if and only if none of its attackers is accepted. For the introduced AF F we obtain the ADF $D_F = (\{a, b, c, d\}, \{\phi_a, \phi_b, \phi_c, \phi_d\})$ where $\phi_a = \top$, $\phi_b = \neg a \wedge \neg d$, $\phi_c = \neg b$ and $\phi_d = \neg c$. We mention that the definitions for semantics in case of ADFs are more involved than in case of AFs since they rely on different (pre-)fixpoints of associated consequence operators (cf. [19] for more information).

We already mentioned that ADFs may express more than *single attacks* as in case of AFs. For instance, the acceptance function $\phi_a = c$ encodes *single support*, i.e. statement a should be accepted, if statement c is. In other words, the acceptance of c leads to the acceptance of a . Another form of support is *collective support*. The acceptance functions $\phi_a = b \wedge c$ is such an example. It encodes that a should be accepted if both b and c are accepted. This means the acceptance of only one of them is not sufficient for supporting a .

Temporal aspects of argumentation: The classical definition of ADFs does not provide one with temporal notions. However, in daily life we are often faced with statements/laws which are valid for a certain time only or depend on the past development, e.g. “You can continue working in the company as long as the Brexit is not delivered”, “From the beginning of next year it will be not allowed to build a nightclub near a residential area” or “I will spend my holidays in France given that I get a salary increase this year.” To encode such expressions, we need to be able to distinguish between different time states related via a certain ordering. We, therefore, introduce so-called *timed Abstract Dialectical Framework* (tADF)³ which are powerful enough to model many frequently occurring temporal restrictions. More precisely, a *timed Abstract Dialectical Framework* (tADF) will be a classical ADF equipped with a countable set T of time states. Moreover, we assume that this set is totally ordered, i.e. there is a binary relation \leq over T which is antisymmetric, transitive and connex. For simplicity, we may assume that T is a subset of the first natural

³ A first version is currently under review for COMMA 2020 (see <https://comma2020.dmi.unipg.it/>).

numbers with the inherited standard ordering. Now, a certain time state n might stand for an hour, a day, a month or whatever granularity is needed. In doing so we are able to speak about the same statement s at different time points t in the future, denoted as s_t . Accordingly, we will have timed acceptance conditions ϕ_{s_t} for any statement s at any time point t . For instance, the condition $\phi_{s_5} = a_1 \vee a_2 \vee a_3 \vee a_4$ encodes a support of s at time point 5 via the statement a for any time point between 1 and 4. If the numbers are interpreted as the first months of the year and if s and a are standing for “I am on vacation in France” or “I have a salary increase”, respectively, then ϕ_{s_5} expresses the statement “I will be vacationing in France in May if I get a salary increase between January and April.”

4 Controlled Natural Languages

As we have seen, computational models of argumentation provide us with a set of tools for logical reasoning about abstract arguments. However, for any useful empirical application, we need instantiations of abstract arguments with real-world arguments as commonly expressed and understood in human communication. Unfortunately, natural language argumentation comprises a large variety to express semantically equivalent statements in heterogeneous ways, and cannot be parsed unambiguously by machines. To facilitate the necessary linking between abstract and real-world arguments, we suggest employing a controlled natural language. In general, a CNL can be seen as a small subset of a natural language [12]. This shrinking can happen in order to increase comprehension or formal precision. For our project, we rely on Attempto Controlled English (ACE), which is one of the most mature CNLs [11]. ACE sentences are notated in a simplified English, which allows a intuitive understanding by human readers. At the same time, each sentence has exactly one logical representation, which prevents ambiguity. For more details see e.g. [11].

For reasoning with ACE two steps are performed. First, a parser translates the ACE sentences into a logical representation. Then, a reasoner is applied to the preprocessed data to get the actual logical inferences. For the first step, the Attempto Parsing Engine (APE) is used, which processes the input into a discourse representation structure (DRS) [10]. The DRS representation can then be further translated, e.g. into a first-order representation, or into semantic web languages like OWL or SWRL, which enables the use of several reasoners [9].

As reasoner we use for the following examples the web interface of RACE (see Fig. 2).⁴ This sophisticated reasoner allows us to check consistency, logical entailment and can

Axioms
 The n:minimum-wage is a n:baseline for all n:salaries. A n:baseline for all n:salaries v:enables a n:living-standard and v:guarantees some n:social-security. The n:minimum-wage is not a n:baseline for all n:salaries.

Check Consistency Prove Answer Query

Check Consistency

overall time: 0.408 sec; RACE time: 0.003 sec

Axioms are **inconsistent**. The following minimal subsets of the axioms cause inconsistency:

- Subset 1
 - 1: The n:minimum-wage is a n:baseline for all n:salaries .
 - 3: The n:minimum-wage is not a n:baseline for all n:salaries .

Fig. 2 Reasoning with RACE over an inconsistent set of ACE statements

be used for query answering. RACE works on the level of logical formulas and does not include the abstract argumentation approach, which we want to use for FAME. The idea of the following examples is to illustrate, how an ACE representation of natural language can be used for logical reasoning and query answering. In addition, possible ideas on how the examples could be modeled with the means of abstract argumentation are provided. In a mid-term perspective, it is planned that the abstract argumentation representation is directly processed by its own customized reasoner framework.

An illustrative example: Let us illustrate the application of our approach with some simplified examples about the introduction of a statutory minimum wage. An argument a_1 that favors the introduction might sound like the following: ‘*Minimum wage is a baseline for salaries. Therefore, it defines a living standard and guarantees social security.*’ Translated into the ACE language this argument might sound like the following: ‘*The minimum-wage is a baseline for all salaries. A baseline for all salaries enables a living-standard and guarantees some social-security.*’ However, this ACE sentence still generates an error during parsing because it uses words unknown to APE. In general, there is the possibility to specify a dictionary for APE or, alternatively, one can specify unknown words with prefixes in RACE directly. For example ‘v:’ is used to denote verbs, and ‘n:’ is used to denote nouns, yielding this modification: ‘*The n:minimum-wage is a n:baseline for all n:salaries. A n:baseline for all n:salaries v:enables a n:living-standard and v:guarantees some n:social-security.*’ Such an argument might be confronted with an argument a_2 expressing ‘*Minimum wage is so low, that it will not change anything.*’. This can be translated into proper ACE as: ‘*The n:minimum-wage is not a n:baseline for all n:salaries.*’

⁴ <http://attempto.ifi.uzh.ch/race/>.

To analyze the given arguments as well as to obtain an abstract argument representation RACE offers several possibilities. First, we may check consistency. If the previous ACE statements are input to RACE, one gets notified that the sentences are inconsistent (see Fig. 2). This can be interpreted as the arguments a_1 and a_2 attacking each other. Moreover, RACE shows a minimal subset causing inconsistency. This means, that already a proper part of a_1 , namely ‘Minimum wage is a baseline for salaries’ (so to speak a sub-argument of a_1) conflicts with a_2 . Moreover, it can be checked that the sentence ‘A baseline for all salaries enables a living-standard and guarantees some social-security.’ is consistent with a_2 . Such information can be used for a more fine-grained perspective on arguments. More precisely, the argument a_1 might be split into two sub-arguments a_1^1 and a_1^2 , and only the latter conflicts with a_2 .

Let us consider another example statement: ‘A minimum wage and therefore higher salaries will only increase automatization or will cost some jobs.’ The intention here is that the *or* in the argument expresses that the person, who uses this argument is a bit unsure about the possible consequences of a minimum wage. Therefore two possibilities are listed which do not have to be true at the same time. Translated to ACE this argument might be represented as: ‘If there is a *n*:minimum-wage then there are some *n*:higher-salaries. Every *n*:higher-salaries *v*:lead to some *n*:automatization or *v*:cost some jobs. There is a *n*:minimum-wage.’ Note that here an extra sentence is added which states the existence of a minimum wage in order to initiate the logical reasoning with the ‘if-then’ part of the first sentence. From an abstract point of view, this statement can be modeled as an argument a_3 resembling a minimum-wage, and the arguments a_4 and a_5 representing the arising of automatization or the loss of jobs, respectively. One suitable way to derive a support relation between arguments is simply logical consequence, i.e. n_1 supports n_2 , if the latter is a consequence of the former. Thus, we may state that a_3 supports a_4 and a_5 given that we include some background knowledge like the general trend of mechanization in the economy which reduces the need for manual labor.

RACE also offers the function query answering. Regarding our example, we may ask ‘Are there some jobs?’. RACE states that this question cannot be answered. The reason for this is that it is not known whether the first or the sec-

ond disjunct is true. Let us assume that a new argument arises saying that there is no automation due to a high machine tax. This means, we add the sentence ‘There is no *n*:automatization.’ Now, RACE states that the initial question can be answered and lists also the minimal subset necessary for this. Consequently, with the help of query answering, we may complete our KB and derive further support relations on the abstract level. Moreover, we may detect missing (implicit) background assumptions to answer a question and provide this information to the user.

5 The Road Map to FAME

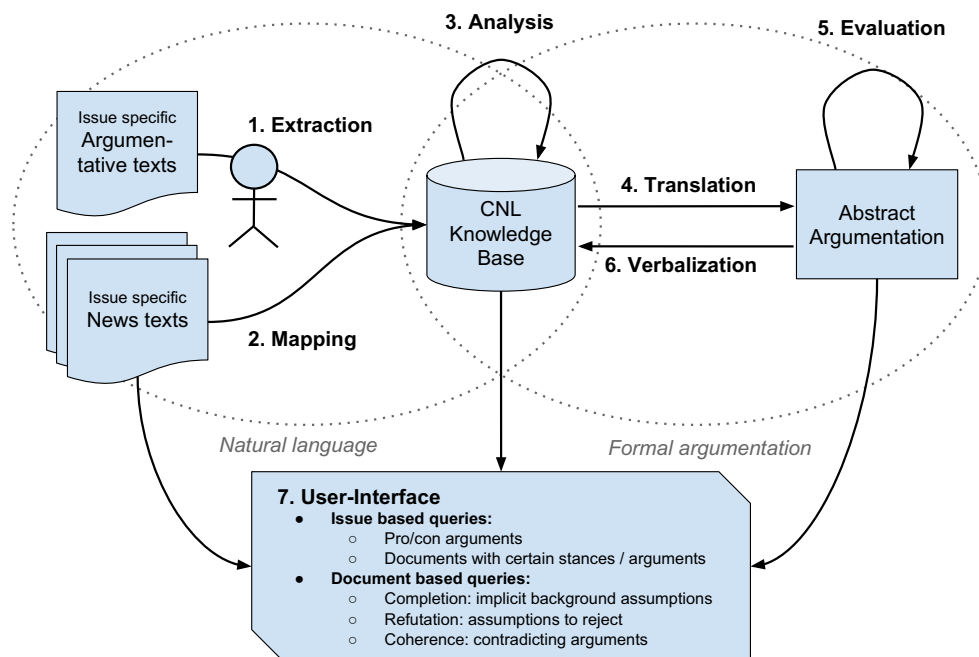
A central goal of the FAME project is to convert argumentation from empirical texts into a formal representation. However, translating arguments from natural language directly to ACE is an AI-complete problem out of the scope of our project. Instead, we will simplify this to a mapping-problem which can be solved in a supervised learning paradigm. On this basis, we strive to answer two research questions in the course of the project: (1) how can we effectively map arguments in empirical text to known ACE statements in a knowledge base, and (2) how can we derive abstract frameworks (e.g. AF, ADF or tADF) from subsets of this knowledge base for automatic evaluation? Fig. 3 presents an overview of the major steps of the project architecture. From the NLP perspective, we start with collecting and extracting issue-specific samples of real-world argumentation from existing resources such as the UKP SAM corpus [18]. Semantically equivalent sets of arguments are grouped together in a computer-aided manner. For this, we employ semantic similarity search technologies based on contextualized word embeddings [21]. Table 1 shows an example of equivalent argument sentences retrieved from the UKP SAM corpus and in line with our previous minimum-wage example.

Accordingly, in step 3 corresponding ACE statements to the retrieved arguments from step 1 are formulated. The resulting KB then needs to be translated into a suitable abstract representation including attack and/or support relations as well as timed arguments if necessary (step 4). In Sect. 4, we sketched how such relations (with the help of RACE) can be found. Moreover, in Sect. 3 we presented suitable formalisms for this endeavor. Step 5 is engaged

Table 1 Example for mapping of ACE statements to natural language arguments retrieved from the UKP SAM corpus

ACE	If there is a <i>n</i> :minimum-wage then there are some <i>n</i> :higher-salaries. Every <i>n</i> :higher-salaries <i>v</i> :lead to some <i>n</i> :automatization or <i>v</i> :cost some jobs.
NL	If the minimum wage is increased, companies may use more robots and automated processes to replace service employees. Ordering businesses to pay entry-level workers more will make them hire fewer of them, and consider replacing more workers with robots or computers. After that, it doesn’t encourage higher wages, and only encourages unemployment and automation.

Fig. 3 Project architecture of FAME



with determining acceptable positions, i.e. acceptable subsets of arguments for a given scenario w.r.t. an argumentation semantics. The main aim of the verbalization step 6 is the retranslation in CNL of the evaluation outcome extending the KB with exactly this information. What do we gain from this? A major goal from the NLP perspective is to develop a precise enough mapping from arguments in empirical text collections such as newspaper articles to arguments in our KB such that we can support various argument analysis and retrieval tasks (7). Potential queries could be ‘Is the argumentation in a given document consistent?’, or ‘What implicit background assumptions does the author assume given the presented arguments in his/her document?’. We believe such a system will allow for many exciting possibilities for domain experts to analyze not only single documents but also argumentation patterns in large document collections. We look forward to sharing our insights, technologies, and resources and with the two argument communities and bringing them more closely together in the near future.

Acknowledgements We thank the DFG (Deutsche Forschungsgemeinschaft) for funding this work (project no. 406289255, funding period: 05/2019–04/2022), and the coordinators of the SPP 1999 for our subsequent admission to the Priority Programme.

Funding Open Access funding provided by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are

included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Bar-Haim R, Bhattacharya I, Dinuzzo F, Saha A, Slonim N (2017) Stance classification of context-dependent claims. In: Proc. ACL, ACL, Valencia, Spain, S 251–261
2. Baumann R (2019) On the existence of characterization logics and fundamental properties of argumentation semantics. Habilitation treatise. Leipzig University, Leipzig <https://doi.org/10.13140/RG.2.2.21831.24483>
3. Besnard P, Hunter A (2008) Elements of Argumentation. MIT Press, Cambridge
4. Besnard P, Garcia A, Hunter A, Modgil S, Prakken H, Simari G, Toni F (2014) Special issue: tutorials on structured argumentation. *Argument Comput* 5(1):1–117. <https://doi.org/10.1080/19462166.2013.869764>
5. Brewka G, Woltran S (2010) Abstract dialectical frameworks. In: Proc. KR, S 102–111
6. Chernodub A, Oliynyk O, Heidenreich P, Bondarenko A, Hagen M, Biemann C, Panchenko A (2019) Targer: neural argument mining at your fingertips. In: Proc. ACL Florence
7. Dung PM (1995) On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif Intell* 77(2):321–357
8. Eckle-Kohler J, Kluge R, Gurevych I (2015) On the role of discourse markers for discriminating claims and premises in argumentative discourse. In: Proc. EMNLP ACL, Lisbon, S 2236–2242 <https://doi.org/10.18653/v1/D15-1267>
9. Fuchs NE (2012) First-order reasoning for Attempto controlled English. In: Rosner M, Fuchs NE (Hrsg) Proc. CNL, Bd. 7175. Springer, Berlin Heidelberg, S 73–94

10. Fuchs NE, Kaljurand K, Kuhn T (2010) Discourse representation structures for ACE 6.6. Tech. Rep. ifi-2010.0010. Department of Informatics, University of Zurich, Zurich
11. Kuhn T (2010) Controlled english for knowledge representation. Ph.D. thesis. Faculty of Economics, Business Administration and Information Technology of the University of Zurich, Zurich
12. Kuhn T (2014) A survey and classification of controlled natural languages. *Comput Linguist* 40(1):121–170. https://doi.org/10.1162/COLI_a_00168
13. Levy R, Gretz S, Sznajder B, Hummel S, Aharonov R, Slonim N (2017) Unsupervised corpus-wide claim detection. In: Proc. Workshop on Argument Mining ACL, Copenhagen, S 79–84 <https://doi.org/10.18653/v1/W17-5110>
14. Palau RM, Moens MF (2009) Argumentation mining: the detection, classification and structure of arguments in text. In: Proc. ICAIL. ACM, New York, S 98–107 <https://doi.org/10.1145/1568234.1568246>
15. Peldszus A, Stede M (2013) From argument diagrams to argumentation mining in texts: a survey. *Int J Cogn Inform Nat Intell* 7(1):1–31. <https://doi.org/10.4018/jcini.2013010101>
16. Reiser P, Inoue N, Okazaki N, Inui K (2015) A computational approach for generating Toulmin model argumentation. In: Proc. Workshop on argumentation mining. ACL, Denver, S 45–55 <https://doi.org/10.3115/v1/W15-0507>
17. Stab C, Gurevych I (2014) Identifying argumentative discourse structures in persuasive essays. In: Proc. EMNLP, S 46–56
18. Stab C, Miller T, Schiller B, Rai P, Gurevych I (2018) Cross-topic argument mining from heterogeneous sources. In: Proc. EMNLP. ACL, Brussels, S 3664–3674 <https://doi.org/10.18653/v1/D18-1402>
19. Strass H (2013) Approximating operators and semantics for abstract dialectical frameworks. *Artif Intell* 205:39–70. <https://doi.org/10.1016/j.artint.2013.09.004>
20. Walton D, Reed C, Macagno F (2008) Argumentation schemes. Cambridge University Press, Cambridge
21. Wiedemann G, Remus S, Chawla A, Biemann C (2019) Does BERT make any sense? Interpretable word sense disambiguation with contextualized embeddings. In: Proc. KONVENS Erlangen, S 161–170