The Formal Argumentation Libraries of Tweety

Matthias Thimm

Institute for Web Science and Technologies (WeST), University of Koblenz-Landau, Germany

Abstract. We provide an overview on the argumentation libraries of the Tweety library collection to artificial intelligence and knowledge representation. These libraries comprise of implementations to abstract argumentation frameworks, as well as the most popular approaches to structured argumentation, and various further aspects. We briefly sketch the functionalities of these libraries and give some pointers to how they can be used.

1 Introduction

The *Tweety libraries for logical aspects of artificial intelligence and knowledge representation*[1] [20, 21] are a comprehensive collection of Java libraries for various logical approaches to artificial intelligence. The Tweety libraries provide implementations of formalisms such as default logic [17], answer set programming [9], belief revision [11], and, in particular, formal argumentation [6, 1, 23, 16, 3, 8, 19, 12, 13, 15, 14].

The popularity of the *International Competition on Computational Models of Argumentation*[2] (ICCMA) has shown that there is a growing interest in algorithmic approaches to formal argumentation. The formal argumentation libraries of Tweety address this by providing a general and versatile collection of Java classes to deal with various aspects of different approaches. The aim of this is not to provide highly efficient implementations, but rather a simple and clear representation of argumentation concepts in an object-oriented manner that can be easily understood and used by researchers and students not trained in algorithm and software engineering.

The remainder of this paper gives a brief overview on the functionalities provided within Tweety for the area of formal argumentation.

2 Overview

Tweety aims at providing a common framework for implementing different approaches to artificial intelligence in general and knowledge representation in particular. It can be used by undergraduate students to better understand logical approaches to knowledge representation by actually working with them in a familiar object-oriented manner. Moreover, the main purpose of Tweety is to allow the easy implementation of new approaches by following a given strict framework and with the benefit of easily integrating concepts and methods of other formalisms. This allows for early testing of ideas and experimental evaluation in terms of feasibility studies.

[1] `http://tweetyproject.org`
[2] `http://argumentationcompetition.org`

Tweety is organized as a modular collection of Java libraries with a clear dependence structure. Each knowledge representation formalism has a dedicated Tweety library which provides implementations for both syntactic and semantic constructs of the given formalism as well as reasoning capabilities. Several libraries provide basic functionalities that can be used in other libraries. Among those is the *Tweety Commons* library which contains abstract classes and interfaces for all kinds of knowledge representation formalisms. Furthermore, the library *Math* contains classes for dealing with mathematical problems such as constraint satisfaction or optimization problems that often occur, in particular, in probabilistic approaches to reasoning. Most other Tweety projects deal with specific approaches to knowledge representation. Each Tweety library is organized as a Maven[3] project. Most libraries can be used right away as they only have dependencies to other Tweety libraries. Some libraries provide bridges to third-party libraries such as numerical optimization solvers which are not automatically found by Maven and have to be installed beforehand. However, all necessary third-party libraries can be installed by executing a single install file located within the Tweety distribution. We refer to [21] for a more detailed description of Tweety in general.

3 Argumentation Libraries

The package `net.sf.tweety.arg` is the general parent package for all approaches pertaining to formal argumentation. In the following, we briefly sketch the functionalities of the sub-package `net.sf.tweety.arg.dung` for abstract argumentation (Section 3.1), various sub-packages for structured argumentation (Section 3.2), and further approaches (Section 3.3).

3.1 Abstract Argumentation

Abstract argumentation frameworks (AAFs) due to Dung [6] are arguably the most investigated formalism for formal argumentation. An AAF is a tuple $AF = (A, R)$ where A is a set of arguments—atomic entities without inner structure—and R is a relation $R \subseteq A \times A$ modelling directed attack between arguments. Thus, an AAF can be represented as a directed graph. Semantics are given to these graphs using *extensions*, i. e. sets of arguments that are jointly acceptable according to some specific acceptance condition [6, 1].

The Tweety package `net.sf.tweety.arg.dung` contains several classes for dealing with AAFs. The class `DungTheory`[4] models an AAF and provides several convenience methods for accessing the data structure and manipulate it. Abstract argumentation frameworks can be imported using the APX format [7] or programmatically using specific methods (see also Figure 1). Tweety supports reasoning with AAFs using the extension-based approaches of grounded, stable, complete, preferred, ideal, semi-stable, CF2, and stage semantics as well as the ranking-based approaches of [10, 22].

[3] `http://maven.apache.org`

[4] The class name `DungTheory` was chosen in favour of the class name `AbstractArgumentationFramework` in order to avoid confusion with the Java term `abstract` which is usually used as a prefix of an abstract class.

```
DungTheory aaf = new DungTheory();
Argument a = new Argument("a"), b = new Argument("b"), c = new Argument("c");
aaf.add(a); aaf.add(b); aaf.add(c);
aaf.add(new Attack(a,b)); aaf.add(new Attack(b,a)); aaf.add(new Attack(b,c));

AbstractExtensionReasoner reasoner = new
                          StableReasoner(aaf, Semantics.CREDULOUS_INFERENCE);

System.out.println(reasoner.getExtensions());
```

Fig. 1. Code snippet for manually creating a simple AAF and determining its stable extensions.

Finally, the package contains an implementation of the logic of dialectical outcomes of [13] that allows modelling and reasoning with extensions of subgraphs, and several factory classes for generating random AAFs.

Figure 1 shows a code snippet for creating a simple AAF and determining its stable extensions.

3.2 Structured Argumentation Approaches

Tweety contains implementations of the most popular approaches to structured argumentation, namely ASPIC$^+$ [16], Assumption-based Argumentation (ABA) [23], Defeasible Logic Programming (DeLP) [8], and *deductive argumentation* [3]. In general, an approach to structured argumentation aims at providing an inner structure to arguments by allowing the representation of those through sets of formulas in some logic. For example, in the framework of *deductive argumentation* [3] classical logic—propositional and first-order logic—is used as the underlying knowledge representation formalism. Arguments are build from classical formulas by identifying a set of classical formulas as the *premise* and a single formula as the *conclusion* of an argument, such that the premise entails the conclusion. Therefore, arguments correspond to minimal proofs in the classic logical sense. If a knowledge base is inconsistent, arguments and counterarguments for different conclusions can be extracted from this knowledge base and put in relation to each other. While [3] bases its framework on classical logic, ASPIC$^+$, ABA, and DeLP also incorporate aspects of non-classical formalisms that allow e. g. the use of default reasoning techniques for the construction of arguments.

Tweety provides several functionalities for importing and working with knowledge bases in ASPIC$^+$, ABA, DeLP, and deductive argumentation. In particular, reasoning with these approaches can be reduced to reasoning with abstract argumentation frameworks by determining the corresponding AAF and using AAF reasoners as discussed above. Note that this is the standard semantical approach for ASPIC$^+$ and ABA. Note, however, that both DeLP and the deductive argumentation approach of [3] also provide proprietary reasoning mechanisms based on the construction of dialectical trees (or *argument graphs* in [3]) and their evaluation. Tweety provides implementation of these reasoning mechanisms as well, in particular the approach through knowledge base compilation for deductive argumentation from [2]. Finally, a web interface for the De-

```
=> WearsRing                  |  AspicArgumentationTheory<PropositionalFormula> t =
=> PartyAnimal                |             parser.parseBeliefBaseFromFile(<File>);
                              |  DungTheory aaf = t.asDungTheory();
d1: WearsRing => Married      |
d2: PartyAnimal => Bachelor   |  AbstractExtensionReasoner reasoner =
                              |                   new PreferredReasoner(aaf,
s1: Married -> ! Bachelor     |                     Semantics.CREDULOUS_INFERENCE);
s2: Bachelor -> ! Married     |  System.out.println(reasoner.getExtensions());
```

Fig. 2. The Tweety format of the classical ASPIC$^+$ example of the bachelor [16] (left) and a code snippet for reading this file into an `AspicArgumentationTheory`, inducing its abstract argumentation framework, and determining the latter's preferred extensions (right).

feasible Logic Programming approach is also available[5] and similar interfaces for other approaches are currently in development.

Figure 2 shows a small example using Tweety's ASPIC$^+$ implementation.

3.3 Further Approaches

Tweety also provides implementations to further approaches to formal argumentation, in particular to various approaches to probabilistic argumentation [19, 12, 13, 15] and how those can be used for opponent modelling in strategies for persuasion [18]. Finally, Tweety provides an implementation of the approach of *social abstract argumentation* [14].

4 Conclusion

We gave a brief overview on the argumentation libraries of Tweety. In particular, we sketched the functionalities of libraries pertaining to abstract argumentation, structured argumentation, and further approaches.

Tweety is an active project and new approaches are added to the collection regularly. Current work is on an implementation for *Abstract Dialectical Frameworks* [5] as well as further approaches to ranking semantics [4].

References

1. Baroni, P., Caminada, M., Giacomin, M.: An introduction to argumentation semantics. The Knowledge Engineering Review 26(4), 365–410 (2011)
2. Besnard, P., Hunter, A.: Knowledgebase compilation for efficient logical argumentation. In: Proceedings of the 10th International Conference on Knowledge Representation (KR'06). pp. 123–133. AAAI Press (2006)
3. Besnard, P., Hunter, A.: Elements of Argumentation. The MIT Press (June 2008)
4. Bonzon, E., Delobelle, J., Konieczny, S., Maudet, N.: A comparative study of ranking-based semantics for abstract argumentation. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI'16) (2016)

[5] `http://tweetyproject.org/w/delp`

5. Brewka, G., Ellmauthaler, S., Strass, H., Wallner, J.P., Woltran, S.: Abstract dialectical frameworks revisited. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13) (2013)
6. Dung, P.M.: On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artificial Intelligence 77(2), 321–358 (1995)
7. Egly, U., Gaggl, S.A., Woltran, S.: ASPARTIX: Implementing Argumentation Frameworks Using Answer-Set Programming. In: de la Banda, M., Pontelli, E. (eds.) Logic Programming, 24th International Conference, ICLP 2008, Udine, Italy, December 9-13 2008, Proceedings. Lecture Notes in Computer Science, vol. 5366, pp. 734–738. Springer (2008)
8. Garcia, A., Simari, G.R.: Defeasible logic programming: An argumentative approach. Theory and Practice of Logic Programming 4(1–2), 95–138 (2004)
9. Gelfond, M., Leone, N.: Logic programming and knowledge representation - the a-prolog perspective. Artificial Intelligence 138(1-2), 3–38 (June 2002)
10. Grossi, D., Modgil, S.: On the graded acceptability of arguments. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15). pp. 868–874 (2015)
11. Hansson, S.O.: A Textbook of Belief Dynamics. Kluwer Academic Publishers, Norwell, MA, USA (2001)
12. Hunter, A., Thimm, M.: On partial information and contradictions in probabilistic abstract argumentation. In: Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR'16) (April 2016)
13. Hunter, A., Thimm, M.: Optimization of dialectical outcomes in dialogical argumentation. International Journal of Approximate Reasoning 78, 73–102 (July 2016)
14. Leite, J., Martins, J.: Social abstract argumentation. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11) (2011)
15. Li, H., Oren, N., Norman, T.J.: Probabilistic argumentation frameworks. In: Proceedings of the First International Workshop on the Theory and Applications of Formal Argumentation (TAFA'11) (2011)
16. Modgil, S., Prakken, H.: The aspic+ framework for structured argumentation: a tutorial. Argument and Computation 5, 31–62 (2014)
17. Reiter, R.: A logic for default reasoning. Artificial Intelligence 13, 81–132 (1980)
18. Rienstra, T., Thimm, M., Oren, N.: Opponent models with uncertainty for strategic argumentation. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13) (August 2013)
19. Thimm, M.: A probabilistic semantics for abstract argumentation. In: Proceedings of the 20th European Conference on Artificial Intelligence (ECAI'12) (August 2012)
20. Thimm, M.: Tweety - A Comprehensive Collection of Java Libraries for Logical Aspects of Artificial Intelligence and Knowledge Representation. In: Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR'14). pp. 528–537 (July 2014)
21. Thimm, M.: The Tweety Library Collection for Logical Aspects of Artificial Intelligence and Knowledge Representation. Künstliche Intelligenz 31(1), 93–97 (March 2017)
22. Thimm, M., Kern-Isberner, G.: On controversiality of arguments and stratified labelings. In: Proceedings of the Fifth International Conference on Computational Models of Argumentation (COMMA'14) (September 2014)
23. Toni, F.: A tutorial on assumption-based argumentation. Argument & Computation 5(1), 89–117 (2014)