Representing Online Debates in the Context of E-Journalism

Elisjana Ymeralli*, Giorgos Flouris*, Vasilis Efthymiou*, Katerina Papantoniou*, Theodore Patkos*,
Georgios Petasis†, Nikiforos Pittaras†, Giannis Roussakis* and Elias Tzortzakakis*
*Institute of Computer Science, FORTH
Heraklion, Greece
Emails: {ymeralli, fgeo, vefthym, papanton, patkos, rousakis, tzortzak}@ics.forth.gr
†Institute of Informatics and Telecommunications, NCSR "Demokritos"
Athens, Greece
Emails: {petasis, pittarasnikif}@iit.demokritos.gr

***Abstract*—An important need of journalists writing opinion articles is the ability to identify, obtain and understand online arguments and opinions covering various perspectives of the debate at hand, a task that cannot be fully accomplished using simple keyword search. This paper describes research on analyzing Greek news articles from a variety of news sources, which considers the internal structure of arguments, in addition to their textual content. We describe a suite of tools for mining, representing and reasoning with real arguments, using semantic technologies, and argue that our tools can enhance future newsroom processes in the domain of online journalism.**

***Keywords*—online journalism; computational argumentation; argument mining; reasoning.**

I. INTRODUCTION

Quality journalism is inherently dependent on reliable and well-justified information delivery, raising the need for automated tools to support journalists in the era of digital information and rapid spread of news feed. One of the crucial components of quality journalism is the ability to identify, search and navigate efficiently in existing journalistic articles to find important arguments related to a topic of interest.

Although standard keyword search is of undeniable value in identifying relevant articles, one cannot use simple keywords to search on the basis of the structure of an argument, or on the basis of its relationships to other arguments. To support these needs, the DebateLab project [1] aims to develop a suite of tools that will allow the journalist, or the interested citizen, to navigate in journalistic articles and understand better their argumentative structure, and, eventually, the main points of both sides of important public debates. Our current focus lies on news articles and sources in the Greek language. However, our approach can easily be extended to other languages.

In this paper, we focus on the underlying representation structure for arguments, which was developed in order to support the above applications. In particular, we will present an ontological model (called *Onto4JARGs - Ontology for Journalistic Arguments* [2]) that we developed to represent, store and reason with arguments, articles and their constituents, in the context of e-journalism, as well as the process that leads to the identification and ingestion of this information in Onto4JARGs.

In the following, we start by presenting the relevant background in the literature (Section II). Section III introduces the main components of Onto4JARGs, used for representing, storing and reasoning with arguments found in journalistic articles. Section IV outlines the pipeline of the ingestion process for identifying and transforming data into ontological information, whereas Section V presents the user-related concepts of Onto4JARGs. We conclude in Section VI.

II. BACKGROUND

A. Related Work

An increasing number of journalistic platforms exist in the literature for harvesting news-related content from the Web, and for analysing and further enriching data with relevant information from various Knowledge Bases (such as Wikipedia). These platforms often use knowledge graphs [3] and other semantic techniques [4] [5] that automatically analyse and enrich news material, and leverage theories and techniques from the field of artificial intelligence [6] and natural language processing [7] [8] to identify, classify, and process news events in a more meaningful way [9].

An example of such a work is described in [10], which describes a prototype for harvesting news-related content and social media messages, using an ontology for representing news items semantically, and well-defined methods from the field of artificial intelligence. Similarly, the *Neptuno* [11] describes an ontology that is useful for semantic search and browsing capabilities, as well as for visualizing content. *NewsReader* [12] [13] is a tool for analysing web news texts semantically and enriching them with relevant information from reference Knowledge Bases in order to built an event-centric knowledge graph. Further, the *EventRegistry* [14] news platform is used to extract data from RSS feeds and link them with relevant information about locations, people and organizations in real time.

Although the aforementioned systems provide useful informative services for journalists, our research is unique in combining many of these features and enriching them

ISBN: 978-1-61208-991-1

with additional functionalities, such as the integration with state-of-the-art ontologies (AIF [15] – Subsection II-B), the identification of argument relationships using computational argumentation methods (Subsection IV-B) and the quantitative evaluation of arguments using sophisticated scoring algorithms (Subsection V-A).

B. Argument Interchange Format (AIF)

The AIF ontology [15] is an abstract ontology for representing argumentation information and relationships among arguments. Its aim is to serve as a "blueprint" towards the definition of more specific and application-dependent ontologies for argumentative information. It is thus closer to a high-level, generic conceptual model for argumentation. Here, we describe the specification used by the Argumentation Research Group at the University of Dundee [16], which is available in various formats [17].

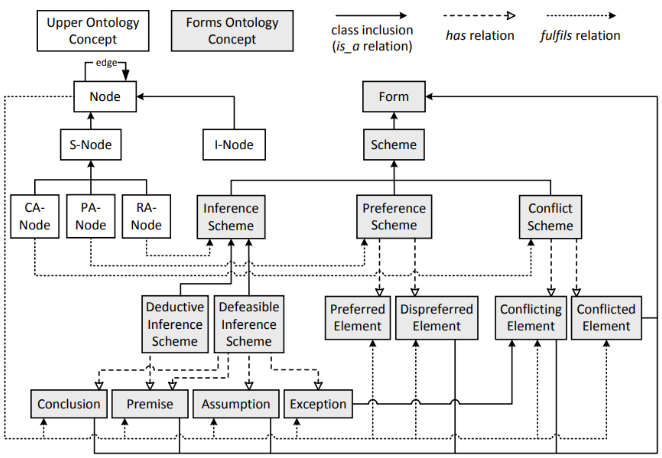

Figure 1. The AIF specification [15].

The main classes of the AIF ontology are shown in Figure 1. AIF consists of two components, namely the Upper Ontology (white boxes) and the Forms Ontology (gray boxes).

The Upper Ontology describes "nodes", which essentially represent arguments and their components. There are two main types of nodes, namely I-Nodes and S-Nodes. Information nodes (I-Node) are used to represent the content of arguments and represent claims and premises that depend on the domain of discourse. Schema nodes (S-Node) represent applications of schemes, i.e., independent patterns of argumentative reasoning. There are three types of S-Nodes, namely CA-Nodes, PA-Nodes and RA-Nodes. CA-Nodes (Conflict Application Nodes) represent conflicts among other nodes, PA-Nodes (Preference Application Nodes) represent preferences, whereas RA-Nodes (Rule of Inference Application Nodes) represent the application of some inference scheme to develop arguments.

The Forms Ontology defines the types of statements and schemes typically used in argumentation. It contains several classes which embody the general principles for actually capturing the pattern of reasoning, which can be an inference rule (*Inference Scheme*), a conflicting rule (*Conflict Scheme*), and a preference rule (*Preference Scheme*). Hence, the individual RA-, CA-, and PA-nodes fulfil these schemes to define whether a conclusion (say c) is inferred, or attacked from the premises (say p_i), as well as denote preferences over nodes.

C. Computational Argumentation

Computational argumentation is the field of study which deals with the representational and reasoning aspects that determine how arguments and argumentative processes can be represented in a computer system, and how the outcome of an argumentative process can be automatically determined [18]. Work in computational argumentation is often classified as either *structured* or *abstract* argumentation.

Structured argumentation concerns itself mainly with the internal structure of an argument, how it should be represented, and how this internal structure determines the relationships between arguments [19] for the representation of domain knowledge. Typically, an argument consists of a set of premises (say Δ) and a conclusion (say c) such that $\Delta \vdash c$, where the $\vdash$ relationship corresponds to the inference relation of the underlying logic. Thus, the argument structure can be viewed as a sequence of statements (i.e., premises) which are often expressed in favor of or against other statements (i.e., given as conclusion), relative to a knowledge graph of structured relations and arguments.

All relations between arguments (e.g., attack, support) are determined by viewing the logical relationships among the argumentative units of the respective arguments, including conclusions and/or premises, as they are understood from the given text (i.e., news article) of the argument mining output. For further explanations on how two or more arguments are related with each other, see also Figure 3 and the related analysis in Subsection III-B.

On the other hand, abstract argumentation ignores the internal structure of arguments and considers only their relationships (e.g., attack, support), attempting to determine the semantics (i.e., the acceptable arguments) given a set of arguments and their relationships. This strand of work was initiated by the work of Dung [20] who viewed a debate as a directed graph, whose nodes are the arguments and the arrows represent attack relationships among them. Since the work of Dung [20], numerous semantics and extensions of the above simple framework have been defined.

III. Main argumentative concepts in Onto4JARGs

For the purposes of DebateLab, we developed *Onto4JARGs* [2], a *Resource Description Framework* (RDF) [21] - based argumentation ontology that fits the needs of the project. Onto4JARGs is heavily based on AIF [15] [16], and essentially enhances and extends its abstract model (Figure 1), reusing most of its main concepts.

Figure 2 shows the main classes and properties of the ontology [2]. In Figure 2, dark blue circles represent classes from the AIF specification, while light blue circles are new classes, introduced by us to represent concepts necessary for

 ISBN: 978-1-61208-991-1

our purposes. Green boxes denote literals that are attributes of the various classes, whose types are indicated using the yellow boxes. In the following, we describe the main components shown in Figure 2, whereas in Section IV we describe the ingestion process that is used to populate the ontology.

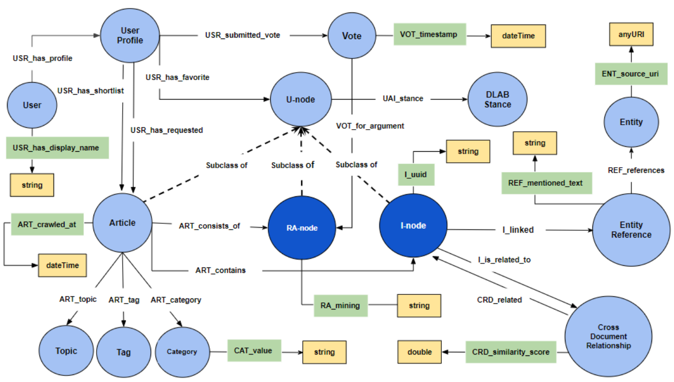

Figure 2. The main components of the Onto4JARGs ontology [2].

A. Arguments and their constituents

In the AIF schema, an argument is considered to be the inference process that reflects reasoning patterns (i.e., argumentation schemes), from which *premises* can be used to derive the *conclusion* of the argument. This inference process (argument) is represented through an RA-node. In Onto4JARGs we follow the same idea and represent arguments using RA-nodes.

We borrow terminology from the argument mining literature and call *Argumentative Discourse Units (ADUs)* [22] the statements that are premises or conclusions of arguments. ADUs can be *major claims, claims* or *premises*, where a major claim constitutes a major conclusion in a journalistic article or document, a claim is an intermediate point in the reasoning process, whereas a premise is a self-evident statement that supports or refutes claims (see Table I for more details). All ADUs are represented using I-nodes, so each RA-node is connected to a set of I-nodes that represent the argument's premises and conclusion.

The CA-nodes are used in AIF to represent conflicts. We use this functionality to represent conflicts among ADUs. These conflicts essentially occur due to the argument generation process (see Subsection IV-B and Figure 5), and are later leveraged to identify arguments that attack each other, using the process described in Subsection III-B and Figure 3.

B. Relations between arguments

The ADUs that comprise an argument, and the relations among them, can be used to determine various types of relationships among arguments, using the process described in [23] [24] and visualised in Figure 3. More specifically, a *rebut* relation occurs whenever a conclusion of an argument conflicts (through a CA-node) with the conclusion of another. An *undercut* relation is identified whenever a conclusion of an argument conflicts with one or more of the premises of another. Analogously, an *endorse* relation occurs whenever a conclusion of an argument is also the conclusion of another, whereas a *backing* relation is identified whenever a conclusion of an argument is also a premise of another. Rebut and undercut are collectively referred to as *attack*, whereas endorse and backing are collectively referred to as *support*.

TABLE I
THE DIFFERENT TYPES OF I-NODES IN THE DEBATELAB ONTOLOGY [22].

ADUs	Description
Major Claim	A major conclusion associated with an article that contains arguments. All arguments in an article are somehow elaborating upon the major claim(s) of the article.
Claim	A statement that can be inferred by or follows as a conclusion of an argument in the article. Each claim is associated with one of the major claims of the article, and may support or refute it.
Premise	A statement within an article that provides a reason for or against some claim.

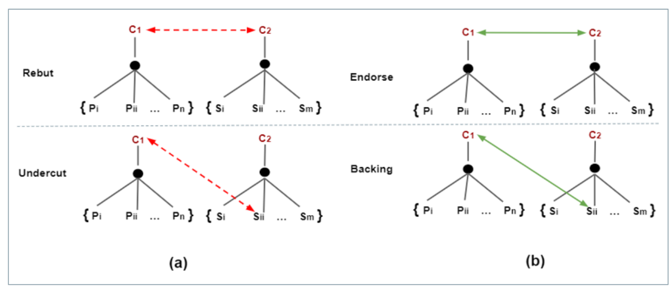

Figure 3. Generation of argument relations using ADU relations [2].

C. The U-node

A central entity in Onto4JARGs is the class U-node, which is used to represent the *Units of Argumentative Information (UAIs)* of DebateLab, i.e., all entities that contain argumentative information (as opposed to metadata, user information and other non-argumentative data). U-node has three sub-classes (e.g., Article, RA-node, and I-node), representing articles, arguments and ADUs, respectively. These sub-classes share common attributes, which are inherited from the U-node class, such as the stance (Subsection IV-C) value and the confidence score (given from the argument mining output), whose role will be explained in subsequent subsections.

IV. THE INGESTION PROCESS POPULATING ONTO4JARGS

Figure 4 provides an overview of the ingestion workflow used in DebateLab. The process starts with the identification of arguments in the text, and the representation of the structured arguments (and their constituents) in an appropriate JSON file (see Subsection IV-A). It continues with the processing of the raw data found in the JSON file in order to generate the various arguments (Subsection IV-B), enrich them with additional information (Subsection IV-C) and finally link them with external data sources (Subsection IV-D). Further, related

 ISBN: 978-1-61208-991-1

arguments and statements from different documents are identified (Subsection IV-E). We analyze these steps in more details in the sequel.

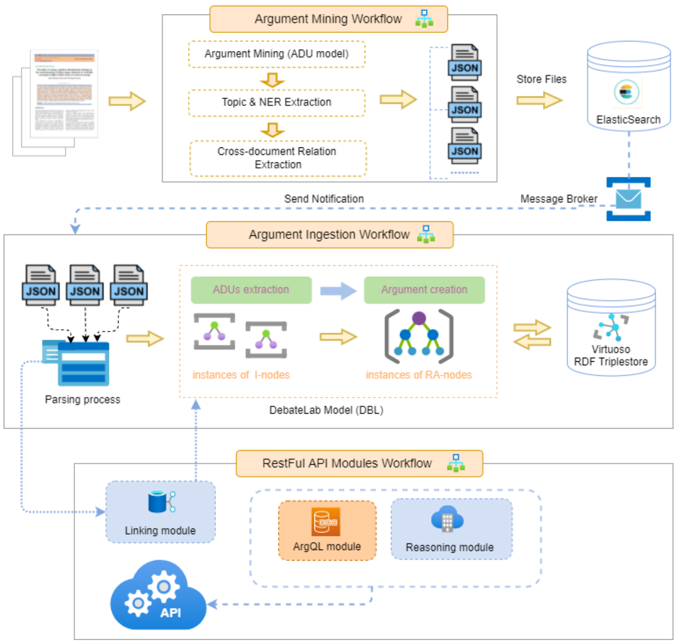

Figure 4. The ingestion workflow of DebateLab [1].

A. News articles detection

The main source of information in DebateLab is journalistic articles crawled from the web, represented through the *Article* class. An article consists of ADUs, organised into arguments, as explained in Subsection III-A. Articles and its constituents (ADUs, arguments) are detected by the argument mining process [25] [26], which is a Natural Language Processing task, aiming to detect and identify argumentative structures from text using Machine Learning methods.

In particular, we utilize transformer models to embed text [27], namely the popular *Bidirectional Encoder Representations from Transformers* (BERT) architecture [28], which defines pre-training schemes on natural language understanding (NLU) tasks such as masked language modeling and next sentence prediction. Due to limited resources in our target language and domain, we take advantage pretrained models [29] in Greek in order to leverage generic knowledge on language structure, prior to fine-tuning on a manually annotated dataset of 150 news articles crawled from the web.

Using these labelled data, we build ADU detection, relation and stance classification models via training token and sequence classification heads on the transformer via supervised fine-tuning. These classifiers yield ADU segments and relation / stance labels on ADU pairs, reaching macro-F1 scores of 0.56, 0.93 and 0.89 respectively in a 3-fold cross-validation setting. Note that, the ADU detector was evaluated using sequence-level rather than token-level matching of predicted and ground truth token sequences. This measure is far harder to satisfy compared to token-level evaluation.

Corresponding baseline macro-F1 scores from uniform random predictions are 0.22, 0.26 and 0.39, indicating that all components perform far above naive baselines and are able to extract argumentative structure from the documents. There is room for improvement in ADU detection, which showcases reduced performance compared to the REL and STANCE components. We estimate that this occurs due to the limited training dataset, label imbalance in the training data as well as the sequence-oriented evaluation. Performance is thus expected to improve for all components as more documents are annotated and added to the training pool.

The process takes as input a set of Greek documents (articles), crawled from the Web, and analyzes them in order to deliver the required information in a structured manner, through appropriate JSON files, which contain the following information regarding an article:

- A list of ADUs classified as premises, claims or major claims.
- The association of each premise to a single claim, and the association of each claim to a major claim. Two types of associations are identified, depending on whether the premise/claim attempts to validate or refute the claim/major claim (denoted by *sup*/*att* respectively).
- A list of metadata (i.e., *identifier, content, confidence score*) connected to each ADU node.
- A set of topics, tags and categories associated with the article.

B. Creating structured arguments

The next step in the ingestion pipeline is to process the raw information provided by argument mining in order to generate arguments and their relationships. The approach we follow for generating structured arguments identifies three different cases (see Figure 5, and the description below), depending on the relationships among the premises and their respective conclusion, as identified by argument mining. Note that for this particular part of the analysis we do not differentiate between claims and major claims, i.e., the argument's conclusion ADU can be either a claim or a major claim.

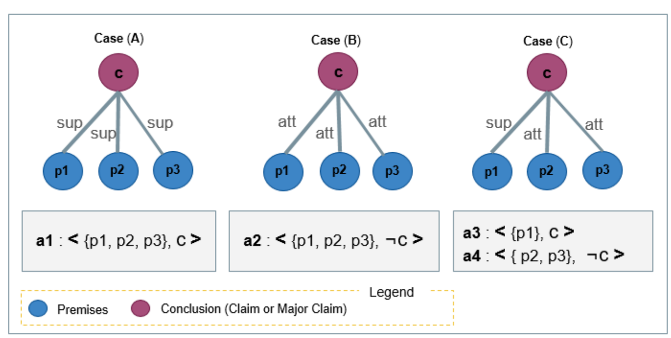

Figure 5. The argument-generation algorithm [2].

Case (A): If all the ADUs (say p_i) associated to a claim or major claim (say c) are associated with it through a *sup* relation, then we create a single argument, whose conclusion

 ISBN: 978-1-61208-991-1

is c and whose premises are all the individual p_i. This case is visualised in the left-most part of Figure 5.

Case (B): If all the ADUs (say p_i) associated to a claim or major claim (say c) are associated with it through an *att* relation, then the process is more complex. First, we create a new, artificial ADU, denoted by $\neg c$, which represents the negation of c. Then, we create a single argument, whose conclusion is $\neg c$ and whose premises are all the individual p_i. This corresponds to the second case of Figure 5. Note that the reason for creating the artificial ADU $\neg c$ is related to the requirement of structured argumentation that the premises of an argument imply the argument's conclusion [19].

Case (C): If a claim or major claim c is associated to different ADUs using both the *att* and *sup* relations, then we apply the above two cases for each group separately. In particular, we generate one argument whose conclusion is c and whose premises are the ADUs associated with c using the *sup* relation (as per Case (A)), and one argument whose conclusion is $\neg c$ and whose premises are the ADUs associated with c using the *att* relation (as per Case (B)). This case is visualised in the right-most part of Figure 5.

C. Enrichment of argumentative data

The *stance* of a UAI represents its attitude (*for* or *against*) towards the topics of an article. To determine the stance, we leverage argument associations. In particular, the ADUs and their associations (*sup/att*) that were identified by the argument mining procedure (Subsection IV-A) in any given article, create a hierarchical structure, whose nodes are the ADUs that appear in the article, and these nodes are connected with the respective *sup/att* associations. By construction, each major claim is a root of a tree (called *ADU tree*), and all the trees created by the major claims together form the *ADU forest* of the article.

Given the ADU forest, the process for determining the stance of ADUs is visualised in Figure 6, and consists of the traversal of each ADU tree in isolation. Specifically, we initialise the stance of all major claims to have the stance value "for". Then, each node (i.e., ADU) in the forest is traversed (root to leaves) to determine its stance as follows:

- If two nodes are connected with the *sup* relation, then the child node inherits the stance value of its parent.
- If two nodes are connected with the *att* relation, then we reverse the stance of the child node, compared to its parent, i.e., if the parent's stance is "for", the child's stance becomes "against" and vice-versa.

D. Entity detection and linking

Our objective in this study is to help a journalist, or an interested citizen, better understand different facets of an issue or debate. Towards this, providing external, objective data and facts about important entities or concepts associated to the various arguments is important. As an example, in a discussion about an important public construction project, data and facts about the project itself, the contractor, or the public body responsible for the construction decisions might be relevant.

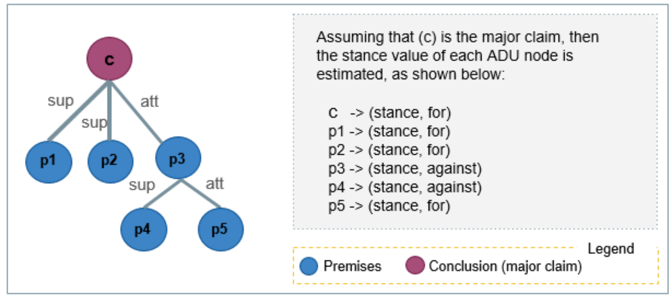

Figure 6. The stance-generation algorithm [2].

To achieve this, the ingestion process incorporates a *linking service* that enriches argumentative data with links to external sources from a variety of datasets from the Linked Open Data cloud (e.g., Wikidata, DBpedia, Wikipedia, etc).

In the ontology, each ADU node is connected to the *Entity Reference* class for identifying relevant named entities that refer to real-world events and associating them with external data sources (through the *Entity* class) using links to related articles, or other items that may help the reader assess their quality and trustworthiness [30]. The current implementation is based on two major steps: (i) extract substrings that are potential entity references, and (ii) identify the external resources (links) that describe the extracted entities. For more details on the following approach and results, see the work in Papantoniou [30].

The implementation process employs state-of-the-art tools, such as BERT [28], wikipedia2vec [31], and fastText [32]. More specifically, a publicly available BERT model is employed for training the *named entity recognizer* (step (i)) for the Greek language [29], while the vector representations of wikipedia2vec, fastText and BERT are used in the disambiguation process, a part of step (ii). The linked entity is selected over possibly many alternatives by calculating the similarity of candidate entity embeddings with the context of the ADU. This approach relies on the *distributional hypothesis* that words with similar meaning are usually found in a similar context.

The output of the linking process is a list of named entities (appearing in the ADUs found in the JSON), coupled with link(s) to related external data sources.

E. Cross-Document relationships detection

Argument mining also allows the identification of cross-document similarity relationships between ADUs appearing in different articles. In particular, each pair of ADUs is associated with a *similarity score*, recorded in the ontology using the class *Cross Document Relationship* and its associated properties. This score is leveraged to compute a similarity score among arguments, allowing users to identify similar arguments and find additional information related to a topic of interest.

V. User-related concepts in Onto4JARGs

In addition to the ingestion process, some parts of the DebateLab database are populated through user actions. These are described in the following subsections.

A. Quantitatively characterising arguments

The need for a quantitative characterization of the arguments' quality, acceptability, or other properties is important for a user who wants to better understand various facets of a debate. To achieve this, we employ the scoring algorithm *s-mDiCE* (*symmetric multi-Dimensional Comment Evaluation*) [33], with the aim to assess arguments along various dimensions (e.g., credibility, quality, acceptance, etc.).

The computation takes into account the relations among arguments, as well as votes that the users have placed on the arguments, represented using the class *Vote*. Each vote affects and is associated with a number of different aspects (such as informativeness, credibility, relevance, etc.) of the argument's evaluation process. A positive vote represents the fact that the user who submitted the vote agrees fully with the content of the argument. In the case of a negative vote, the user needs to specify the reasons he/she disagrees with the content of the argument, by choosing one or more aspects. This is similar to the approach used in other works (e.g., APOPSIS [34]).

Note that some of the evaluated aspects are static and are calculated once, whereas others are dynamically changing. The initial computation of scores takes place during the ingestion process, and recomputation is automatically performed when a relevant action takes place, i.e., a vote is placed by a user, or a new related argument is ingested.

B. Representing user profiles and activities

The *User* entity is used to represent user-related information for registered users interacting with the Debatelab ecosystem. Each user can edit his/her profile, which includes personal information, such as his/her display name (*USR_has_display_name*), year of birth (*UPR_year_of_birth*), registration date (*USR_registration_date*), and others.

In addition, the User entity records information related to the user's interaction with the system and personal preferences. More specifically, users can vote on arguments, directly affecting their evaluation (see Subsection V-A), and such votes are recorded using the *USR_submitted_argument_vote* property. In addition, each user can mark an article as a *favorite*, for easy access in the future (like a standard bookmarking service). Such articles are recorded using the *USR_has_favorite* property. A user can add and remove articles from his/her list of favorites, and can manage and organise this list, through the *Article Archiver* service of DebateLab. Furthermore, the user is allowed to request the ingestion of a new article in the DebateLab database, through the *on-demand article crawler* service. This has the effect of enriching the database with articles that the default crawling service has missed. Last but not least, the user can submit his/her own user-generated arguments through the Enhanced Debate Portal tool (similar to APOPSIS [34]), a debating platform for analysing structured opinions, integrated in the DebateLab ecosystem.

VI. Conclusion

The DebateLab project [1] aims to assist the professional journalist, as well as the interested citizen, to identify, understand, analyse and navigate through arguments appearing in Greek journalistic articles, crawled from the Web. This can help support a better understanding of public debates, and to improved citizenship and e-democracy.

In this paper, we presented the knowledge model of DebateLab, i.e., our approach for representing and storing real arguments, extracted from existing sources (articles) on the Web. The representation approach is based on an ontology, called *Onto4JARGs*, that we built using Semantic Web Technologies based on the well-established AIF ontology [15]. We described the ontology implementation and its main components, the ingestion process (i.e., the process that transforms the textual raw data to structured knowledge represented in the ontology), and the allowed user actions that affect the contents of the DebateLab database.

Our ontology will be used in different scenarios for serving and helping the professional journalist in carrying out his/her daily activities more efficiently, through the implementation of DebateLab tools. Although our work is tailored for use in the DebateLab project (and the respective journalistic use case), we hope that it will be suitable for other scenarios and domains where argumentative information is relevant.

Acknowledgment

This work was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "1st Call for H.F.R.I. Research Projects to support Faculty Members and Researchers and the procurement of high-cost research equipment" grant (Project Number: 4195).

References

[1] FORTH-ICS, "Debatelab project." [Online]. Available: https://debatelab.ics.forth.gr/, 2022, accessed: 2022-08-28.

[2] ONTO4JARGs, "An rdf-based argumentation ontology for supporting journalistic arguments," [Online]. Available: https://isl.ics.forth.gr/debatelab_ontology/, 2022, accessed: 2022-08-28.

[3] A. Singhal, "Introducing the knowledge graph: things, not strings," [Online]. Available: https://blog.google/products/search/introducing-knowledge-graph-things-not/, p. 16, 2012, accessed: 2022-08-28.

[4] D. Allemang and H. James, *Semantic web for the working ontologist: effective modeling in RDFS and OWL*. San Francisco, CA, USA: Elsevier, 2011, ISBN 978-0-12-385965-5.

[5] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American Magazine, USA*, vol. 284, no. 5, pp. 34–43, 2001.

[6] S. Russel and P. Norvig, *Artificial intelligence: a modern approach*. Pearson Education Limited London, 2013, ISBN 978-0136042594.

[7] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *proceedings of the 1st International Workshop on Learning Representations (ICLR)*, Scottsdale, AZ, USA, 2013.

[8] T. Mitchell *et al.*, "Never-ending learning," *Communications of the ACM*, vol. 61, no. 5, pp. 103–115, 2018.

[9] N. L. Latar, "The robot journalist in the age of social physics: The end of human journalism?" in *The new world of transitioned media*. IDC, Herzliya, Israel: Springer, 2015, pp. 65–80, ISBN 978-3-319-09008-5.

[10] A. Berven, O. A. Christensen, S. Moldeklev, A. L. Opdahl, and K. J. Villanger, "A knowledge-graph platform for newsrooms," *Computers in Industry*, vol. 123, p. 103321, 2020.

[11] P. Castells, F. Perdrix, E. Pulido, M. Rico, R. Benjamins, J. Contreras, and J. Lorés, "Neptuno: Semantic web technologies for a digital newspaper archive," in *The Semantic Web: Research and Applications (ESWC)*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 445–458, ISBN 978-3-540-25956-5.

[12] P. Vossen *et al.*, “Newsreader: Using knowledge resources in a cross-lingual reading machine to generate more knowledge from massive streams of news,” *Knowledge-Based Systems*, vol. 110, pp. 60–85, 2016.

[13] M. Rospocher *et al.*, “Building event-centric knowledge graphs from news,” *Journal of Web Semantics*, vol. 37-38, pp. 132–151, 2016.

[14] G. Leban, B. Fortuna, J. Brank, and M. Grobelnik, “Event registry: learning about world events from news,” in *Proceedings of the 23rd International Conference on World Wide Web*, Seoul Korea, 2014, pp. 107–110, ISBN 978-1-4503-2745-9.

[15] C. Chesnevar *et al.*, “Towards an argument interchange format,” *The knowledge engineering review*, vol. 21, no. 4, pp. 293–316, 2006.

[16] C. Chesnevar *et al.*, “The Argument Interchange Format (AIF) Specification,” *Knowledge Engineering Review (KER)*, p. 1–25, 01 2007.

[17] University of Dundee, [Online]. AIF model. Available: http://www.arg.dundee.ac.uk/aif, 2022, accessed: 2022-08-28.

[18] P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, *Handbook of Formal Argumentation*. College Publications, 2018, ISBN 978-1848902756.

[19] Ph. Besnard and A. Hunter, “A review of argumentation based on deductive arguments,” in *Handbook of Formal Argumentation*, P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, Eds. College Publications, 2018, pp. 435–482, ISBN 978-1848902756.

[20] P. M. Dung, “On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games,” *Artificial Intelligence*, vol. 77, no. 2, pp. 321–357, September 1995.

[21] RDF, “Resource description framework,” [Online]. Available: https://www.w3.org/RDF/, 2022, accessed: 2022-08-28.

[22] A. Peldszus and M. Stede, “From argument diagrams to argumentation mining in texts: A survey,” *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI), USA*, vol. 7, no. 1, pp. 1–31, 2013.

[23] D. Zografistou, G. Flouris, and D. Plexousakis, “Argql: A declarative language for querying argumentative dialogues,” in *RuleML+RR*. London, UK: Springer, 2017, pp. 230–237, ISBN 978-3-319-61251-5.

[24] D. Zografistou, G. Flouris, T. Patkos, and D. Plexousakis, “Implementing the argql query,” in *Computational Models of Argument - Proceedings of COMMA*, vol. 305. Warsaw, Poland: IOS Press, 2018, pp. 241–248.

[25] J. Lawrence and C. Reed, “Argument mining: A survey,” *Computational Linguistics, Cambridge, USA*, vol. 45, no. 4, pp. 765–818, 2020, ISBN 0891-2017.

[26] R. M. Palau and M. F. Moens, “Argumentation mining: the detection, classification and structure of arguments in text,” in *Proc. of the 12th ICAIL*, Barcelona, Spain, 2009, pp. 98–107, ISBN 978-1-60558-597-0.

[27] A. Vaswani *et al.*, “Attention is all you need,” in *Advances in neural information processing systems (NIPS)*, vol. 30, CA, USA, 2017, p. 6000–6010.

[28] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” *Proceedings of the 2019 Conference of the Association for Computational Linguistics: Human Language Technologies (NAACL), Minneapolis, Minnesota, USA*, vol. 1, pp. 4171–4186, 2019.

[29] J. Koutsikakis, I. Chalkidis, P. Malakasiotis, and I. Androutsopoulos, “Greek-bert: The greeks visiting sesame street,” in *11th Hellenic Conference on Artificial Intelligence (SETN)*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 110–117, ISBN 978-1-4503-8878-8.

[30] K. Papantoniou, V. Efthymiou, and G. Flouris, “El-nel: Entity linking for greek news articles,” in *Proceedings of the ISWC*, vol. 2980. Virtual Conference: CEUR-WS.org, 2021.

[31] I. Yamada *et al.*, “Wikipedia2vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from wikipedia,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*. [Online]: Association for Computational Linguistics, 2020, p. 23–30.

[32] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” *Transactions of the Association for Computational Linguistics (TACL), Vancouver, Canada*, vol. 5, pp. 135–146, 2017.

[33] T. Patkos, G. Flouris, and A. Bikakis, “Symmetric multi-aspect evaluation of comments extended abstract,” in *In Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI)*. Amsterdam, Netherlands: IOS Press, 2016, pp. 1672–1673, ISBN 978-1-61499-671-2.

[34] E. Ymeralli, G. Flouris, T. Patkos, and D. Plexousakis, “Apopsis: A web-based platform for the analysis of structured dialogues,” in *The 16th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE)*. Rodos, Greece: Springer, 2017, pp. 224–241, ISBN 978-3-319-69458-0.