Argumentative Discourse Concepts as Revealed by Traversing a Graph

Panagiotis Gourgaris, Andreas Kanavos, Nikos Karacapilidis, Vassilis Tampakas

Argumentative Discourse Concepts as revealed by Traversing a Graph

Panagiotis Gourgaris[1], Andreas Kanavos[1,3], Nikos Karacapilidis[2] and Vassilis Tampakas[3]

1. Computer Engineering and Informatics Department
University of Patras, Patras, Greece
{gourgaris, kanavos}@ceid.upatras.gr
2. Department of Mechanical Engineering and Aeronautics
University of Patras, Patras, Greece
karacap@upatras.gr
3. Computer and Informatics Engineering Department
Technological Educational Institute of Western Greece, Antirrion, Greece
vtampakas@teimes.gr

Abstract. Recent advances in computing and Internet technologies, together with the advent of the Web 2.0 era have resulted in the development of a plethora of online tools, such as forums and social networking applications, which offer people an unprecedented level of flexibility and convenience to participate in complex argumentative discourses of diverse interest. However, these tools do not enable an intelligent analysis of the related content. Aiming to address this issue, this paper presents preliminary work on the exploitation of Neo4j graph platform for managing well-established argumentation elements. The proposed high level and scalable approach facilitates the discovery of latent and arbitrarily long and complex argumentation in argument graphs, as well as its meaningful exploitation towards gaining insights.

Keywords: Argumentation, Collaborative Decision Making, Graph Database, Knowledge Graph, NoSQL

1 Introduction

Argumentation is ubiquitous in our everyday life. It might be argued, that every action can be modeled, up to an extent, as an argumentative discourse [13]. An array of examples of various importance and complexity can be pointed out: political and rhetoric argumentation, business negotiations, as well as questions such as "which movie shall we watch tonight", "which car should I buy", and so forth. All these are manifestations of an argumentation discourse.

This paper presents preliminary work on the exploitation of Neo4j for managing argumentation related data. Neo4j, a NoSQL graph database, stores data physically as a graph. Starting here from the notions portrayed in Dung's works [1], [4], [5], [6], we try to map the various argumentation elements and their relations to a graph which is dynamically constructed during a discourse. Taking into account diverse argumentation concepts and approaches, our formalization elements are those described in the Hermes system [10]. Although here, instead of a tree, we loosen up the architecture and consider the whole issue formulation problem as a continuously expanding graph, following the generally accepted notions in argumentation. Given a knowledge graph formulated as an argumentation framework, we examine the possibility of a graph theoretic route that would yield some quantitative results.

Neo4j supports Cypher, a declarative, ASCII art, and pattern based query language for handling conceptual graphs. We try and relax the restrictions imposed by Dung's model for argument sets, in hope of finding a way to quantitatively assess the process of collaboratively formulating the understanding of an issue and reaching consensus. We aim to check whether the relaxed argumentation forms that we will use give a more probabilistic nature to argumentation acceptance and the assent of any issue considered. By including the human factor, our approach by design engulfs notions that we see in the value-based argumentation frameworks propositions and the labeling process for examining argument attacks.

1.1 Motivation & Contribution

Although the overall argumentation process does not necessarily consider large data sets, the rapid development of web technologies and social networks have resulted in the creation of a big volume of content. Opinions, emotions, ideas and thoughts can be extracted in order to explain, assist, and even predict human decision making. The Big data challenge lies in the efficient implementation of a framework for extracting large amount of data from the social media and adopt appropriate methodologies to process them. We can succeed in this by employing a Neo4j server and populating a graph database. The need for new database approaches, that goes beyond the relational one, was highlighted with the advent of Web 2.0, which is dominated by high volumes of unstructured or semi-structured and high order data. Those data can provide the knowledge on which computer formulated notions from argumentation theory like dialogues, value-based argumentation frameworks, argument schemes etc. can rely and give solutions.

The most commonly used tools for knowledge sharing cannot be considered deliberation tools. The problems they face are systemic, like signal-to-noise ratio, redundancy, repetition and balkanization. The main result is not having any debate and low quality of knowledge being spread, although one could contend that this was not their initial goal.

The remaining of this paper is organized as follows. Section 2 presents existing research and background topics on argumentation notions. Section 3 provides a description of Neo4j's basic elements. Section 4 discusses the proposed approach about how the knowledge graph should be created during the deliberation. Section 5 gives a road-map of challenges and comments on limitations of deliberation systems and Graph Databases. Finally, Section 6 provides concluding remarks and sketches our future work directions.

2 Argumentation Concepts

In this section we provide a description of some basic argumentation notions which sparked our study. Their direct relation with graph databases is what made us try and devise a deliberation schema based on a graph database back-end.

To start with, an *argumentation discourse* as a process towards reaching consent over a difference of opinions, has four *discussion stages* as portrayed in [7]:

1. **Confrontation Stage**: this is where doubt or the difference of opinions over a standpoint is brought forward.
2. **Opening Stage**: the departing point for the discussion. A common ground for a fruitful discussion must be reached and all parties, or protagonists, present their own assent towards a standpoint.
3. **Argumentation Stage**: the main discourse stage, where protagonists bring forth arguments to overcome any antagonists' doubts or critical reactions and antagonists are considering acceptance of the protagonists' argumentation.
4. **Concluding Stage**: at this stage, either a protagonist's standpoint is accepted by all parties, hence the antagonists' doubts are retracted, or a protagonist's standpoint is retracted. In any case, a conclusion must be reached and explicitly expressed.

Although these stages seem to have a logical ordering, a discourse need not pass through all of them -at least not explicitly- and not in this particular order. These stages could be easily matched with the diverse processes of deliberation systems; for example, threads in internet forums.

A finer structuring of the elements making up argumentation is enabled through Pollock's theory [14] for defeasible reasoning. Reasons - called arguments, by argumentation theorists - can be attacked by others, hence the use of the adjective defeasible. The main issue risen is when the argument is attacked. Or, in other words, in which way is it attacked? Pollock argues that two types of defeaters exist, namely *rebutting* and *undercutting*.

1. **Undercutting argument**: those attacking the connection of an argument and its conclusion. This is knowledge obtained that attacks the acceptance of the reason supporting a conclusion.
2. **Rebutted**: a reason supporting the opposite conclusion.

The work done by Dung [4] is where argumentation relates most with graph theory. Notions as *argumentation framework*, *conflict*, *relation* are explained in his paper. Specifically:

An *argumentation framework* is a pair

$$AF = \langle A, attacks \rangle$$

where A is a set of arguments and *attacks* is a binary relation defined on A. A framework can be represented as a directed graph $G = (V, E)$. A graph G

consists of a set of vertices V and a set of edges E. Just like in an argumentation framework, the set of edges correspond to a set of pairs of vertices by means of a relation. Hence, for a set $A = \{a, b, c\}$, (a, b) denotes *a attacks b*.

As a starting point for a deliberation system in which a discourse can be unravelled, we use notions from Hermes [10]. Its framework is based on IBIS (Issue-Based Information System) . The argument diagramming takes form through a framework consisting of *issues*, *alternatives*, *positions*, and *constraints*.

1. **Issues**: the problem to be solved.
2. **Alternatives**: the possible solutions to the problem.
3. **Positions**: the arguments as seen in Dung; they support or attack an alternative.
4. **Constraints**: a way to quantitatively weigh reasons; they have the form of a rule $(position_1, preference\_relation, position_2)$ with relations taking a value of *"more(less) important than"* or *"equally important"*.

3 Neo4j Graph Database

Our work aims to build innovative tools that give an overall view of data-intensive and cognitively-complex discourses, while also efficiently handle the diversity of requirements concerning their analysis and meaningful interpretation. By having such a roadmap, one can assess all dialectical rules and constructs. Fallacies and impediments which contradict the whole process can be more evident; and in the end, reaching a solution regarding the issues at hand [17], hopefully could become a lot easier. By using a graph database, one can store relationships between data. This unique characteristic makes them ideal for our purposes.

To our knowledge, Neo4j has not been used for such an endeavor. Examples of use cases include real-time recommendation systems, social network analysis, network and IT operations. It provides production grade front- or back-end social graph storage. Moreover, it offers graph analytics similar to link prediction, shortest paths, clustering coefficient, and minimum spanning trees, bolstering the potential of graph tools including NetworkX, machine learning frameworks like Graphlab, and distributed processing systems such as Spark [12], [16]. In a context where (i) data are highly connected amongst them, (ii) relationships are often created, erased and updated, and (iii) relationships are the elements that actually trigger insights, graph databases are the solution to handling the big data as well as the real-time aspect of today's deliberation systems.

The fundamental units that form a graph are *nodes* and *relationships*. In Neo4j, both can have properties. Entities of a domain are represented by nodes, albeit relationships can be used too, depending on the formulation and domain of the problem. In addition, multiple *labels* can be attached to nodes. The nature of the questions we want to answer, based on our data, will define the various structures of our graph.

Neo4j's data model is similar to the entity relationship diagram. Concretely, *nodes* can be considered as the entities in the graph. So, for example in the statement *"Panos and Andreas are friends and Panos owns a Brand1 car with plate number AF-101 and Andreas owns a Brand2 car with plate number BD-101"*, Panos, Andreas and the cars are all distinct entities, hence will be distinct nodes in our graph.

Labels state the role of a node making them essentially the entity type identifiers. As an excellent paradigm for the statement above, we can locate two labels for nodes, *person* and *car*.

Relationships are any interactions between nodes, usually identified as verbs. They are considered to be directed and have a start node and an end node. In our example statement, those would be *friend of* and *owns*.

Properties are attributes of nodes and relationships. They are *key-value* pairs attached to them, helping us to quantitatively answer questions regarding our database. In these settings, the "Panos" node which is labeled as a *person*, can have a set of properties like $name : Panos, age : 35$. Also the relationship *friend of* can have a pair of *since:MM/DD/YYYY* as its property.

Again, we can easily see that depending on the problem (or domain), the same statement might yield a different data model. For example, although in our previous analysis the plate numbers could simply be a property of nodes labeled as *car*, in another approach specific license plates could be nodes themselves.

The way we put all this to use is through the Cypher language, which was built grounded on the simple SQL clauses and has lots of graph domain additional ones. A simple occurence for our little dataset above could be:

```
match (p:Person)-[:FRIEND]->(f:Person)
where p.name='Panos'
return f.name
```

which would return all names of nodes that the Panos node has a relationship of $FRIEND$. The basic idea is that initially one forms a math statement which will return a subgraph and then applies an action on it. Another example is:

```
match ()-[:FRIEND]->()<-[:FRIEND]-(f)
return f.name
```

which would return the names of all nodes that have a relationship of type $FRIEND$ with some common node. Considering this from an argumentative perspective, these could be two arguments *a,c* attacking the same argument *b*, which could constitute an "accrual of reasons".

4 Proposed Schema

In this section, we present our approach for a schema that could work as a knowledge graph for a Collaborative Decision Making system. We will be using argumentation elements based on those of Hermes [10], in following put them in a Neo4j's context, and finally try to address the problem formulation task. We will be presenting the elements and notions using graph lingo.

Our scheme has three types of nodes and four types of relationships. All relationships (edges) in Neo4j are directed. Incidentally, if a node u points to a node v, then u is a child of v and v is a parent of u.

4.1 Nodes

The types of nodes used in the present schema are *issues*, *alternatives* and *arguments*.

Issues, here noted as S_j, are equivalent to *standpoints* (the problem being debated). In Hermes [10], they refer to decisions to be made; in this paper, we prefer keeping a more argumentative perspective for notations, as this mental disposition helps with the rest of our descriptions. Issues can also be parents for other issues.

Alternatives, here noted as C_i, are children nodes of *issues*. An *alternative* is a proposition, considered to be a potential solution to the parent issue.

Arguments can be children of both *alternatives* and *issues*. Depending on the relationship between two arguments, a new issue can be automatically raised.

4.2 Relationships

We have four types of relationships depending on their start and end node.

Relationship H between an alternative C_i and an issue S_j:

$$C_i\ H\, S_j,\ with\ H\ \epsilon\ \{ALTERNATIVE\_TO\}$$

Relationship Ω between an argument a or an issue S_k and an issue S_j:

$$\{a, S_i\}\ \Omega S_j,\ with\ \Omega\ \epsilon\ \{RAISES\}$$

Relationship Υ between an argument a and another one b:

$$a\ Y\ b,\ with\ \Upsilon\ \epsilon\ \{REBUTS, UNDERCUTS\}$$

Relationship X between an argument a and an alternative C_i:

$$a\ X\ C_j,\ with\ X\ \epsilon\ \{ATTACK, SUPPORT\}$$

Some characteristics of the above relationships are:

1. All relationships are non symmetric.
2. A *rebuttal* automatically raises a new issue. We consider it to be a new issue and view it as a new dispute that needs to be resolved through negotiations.
3. Also, when an argument a rebuts another b, then the two must have opposing relationships with their common alternative parent. So:

   $$if(a, ATTACK, Ci)and(a, REBUTS, b), thenwemusthave(b, SUPPORT, Ci).$$

4. Upon the creation of a rebutting relationship, a new issue S_j concerning it as well as the relationship $(a, RAISES, S_j)$ is created.

4.3 Example

Figure 1 illustrates the main Neo4j web interface. The graph depicted in Figure 1 is more clearly shown in following Figure 2. There, a draft example of a simple argumentation over the question *"Should I buy car Model1 of Brand1?"* is considered.

In addition, two alternatives are presented, namely *"Yes"* and *"No"*. These two alternatives often correspond to two different opinions. Another user supports *"Yes"* with an argument *"Low price compared to others"*. After that, an argument rebutting it (*"Very expensive for what is provides"*) is expressed and a new issue with the title *"Price comparison"* is created. Likewise, we see that the argument *"Engines of Model1 from Brand1 burn oil"* supports *"No"*, while another, arguing *"Engines of Model1 since* 2008 *are fixed"*, undercuts it.

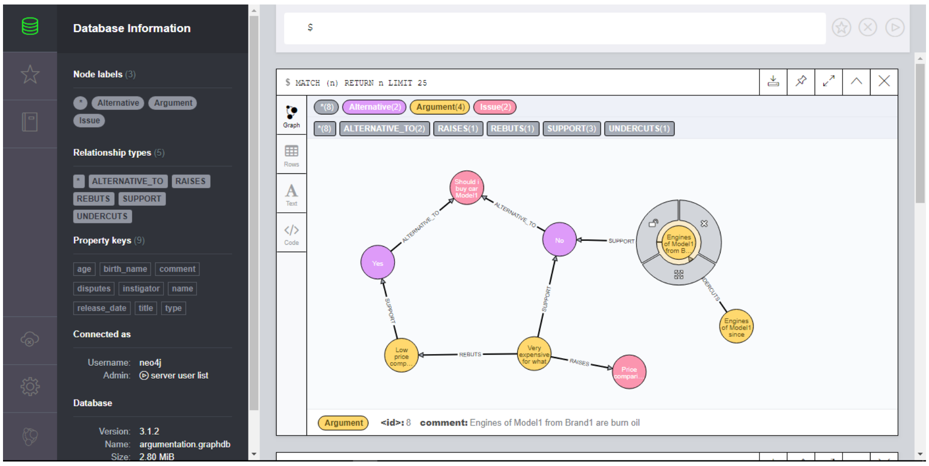

Fig. 1. The web interface of Neo4j administrator panel.

While not fully visible in this simple example, we will be forming our argumentation map with the dialectic systems in mind. Although the time aspect of time-centric systems might be considered by others to generate non-structured data, it is the authors' opinions that it could be an asset, adding to the knowledge that can be obtained from the graph.

5 Roadmap

Towards implementing a deliberation system having Neo4j as its backbone, numerous problems and issues need to be addressed. In such systems, one has to bear in mind the human factor as well as the mechanics of how data are produced.

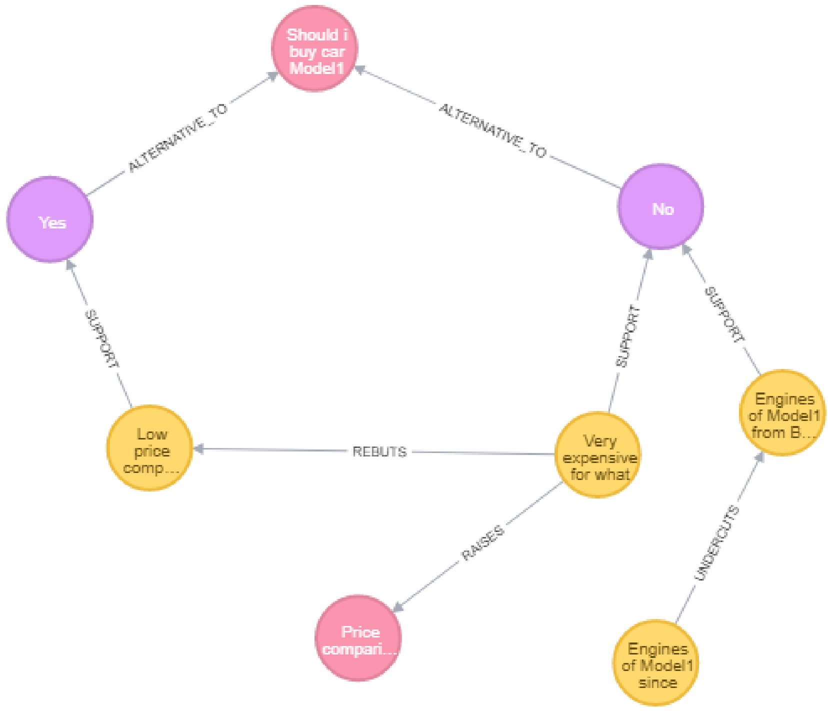

Fig. 2. A simple example of deliberation.

The World Wide Web can be seen as an ideal platform for enhancing argumentative communication and collaboration, due to its ubiquity and openness. Personal blogs and unstructured or semi-structured online discussion forums can provide a medium for such communication.

Notwithstanding opinions and discussions may be identified by their topics, time, or participants, there is a lack of fine-grained structure that captures the way that different facts, opinions, and arguments relate to one another and, as such, contribute to the overall picture. An example is considered in [3], where Deme is specifically designed for supporting democratic, small to medium-sized group deliberation.

By far, the most used systems are those that formulate the deliberation process via sharing content also known as time-centric systems, e.g. wikis, blogs, forums and emails. Data is most commonly in the form of free text and the way people create content does not give a structured result. The size and creation processes of data are the main reasons of problems such as duplication, balkanization as well as the very low signal-to-noise ratio. Also, the only two valid ways of handling the plethora of data and any redundancy as a result, is to either impose restrictions prior to making any post publicly available -perhaps if designated members moderate discussions, or to manually assess all data after the knowledge database is created; in either way, the amount of human labor can become enormous.

A number of highly-structured argument-based deliberation support systems (ADSS) have been already proposed. These suffer from two key limitations: first, they usually support a small number of participants; in addition, most of them target specific domains, such as education, or academic research. Moreover, highly-structured ADSSs based on client-server architectures, are usually designed for small to medium-sized groups and are therefore not easily scalable [8].

Another limitation of existing structured ADSSs is that they subscribe to specific theories of argumentation and decision-making. While the majority of these systems may be suitable for specific domains, an outstandingly truly global-scale argumentation infrastructure must allow for a variety of reasoning patterns to spark interaction. Such reasoning patterns are known in argumentation theory as argumentation schemes [18]. By incorporating such methodologies, the non-collaborativeness problem with large scale tools [11] can be efficiently addressed.

Authors in [15] present theoretical and software foundations for a World Wide Argument Web (WWAW). Concretely, WWAW can be considered as a large-scale Web of inter-connected arguments posted by individuals to express their opinions in a rather structured manner. ArgDF, a pilot Semantic Web-based system is presented, through which users can create arguments using different argumentation schemes and can query arguments using a Semantic Web query language.

6 Conclusions & Future Work

We have presented preliminary work on the exploitation of Neo4j for managing argumentation related data. As future work, we intend to enhance the framework in order to perform meaningful numeric evaluations. More specifically, we will examine whether all types of nodes can have such capabilities. Furthermore, we will try to address several issues that deliberation systems face; heavy dependency on manual moderating work -either during or after the system launch - has to be thoroughly tackled. A very good example regarding this issue is the user incentives of Stack Overflow[1]. In addition to reputation votes, there are about 90 badges a user wins for completing milestones.

Another addition to our framework concerns the argumentation structures [2]. This could help arguments being posted in order to be regarded more as evidence rather than as simple bias, thus adding to the overall credibility of the system. The ultimate goal after dealing with these issues would be the development of an online argumentation platform for generic, large-scale deliberating purposes [9].

[1] `https://stackoverflow.com/`

References

1. Bondarenko, A., Dung, P.M., Kowalski, R.A., Toni, F.: An abstract, argumentation-theoretic approach to default reasoning. Artificial Intelligence 93, 63–101 (1997)
2. Craven, R., Toni, F.: Argument graphs and assumption-based argumentation. Artificial Intelligence 233, 1–59 (2016)
3. Davies, T., O'Connor, B., Cochran, A., Effrat, J.J., Parker, A., Newman, B., Tam, A.: An online environment for democratic deliberation: Motivations, principles, and design. CoRR abs/1302.3912 (2013)
4. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. Artificial Intelligence 77(2), 321–357 (1995)
5. Dung, P.M., Kowalski, R.A., Toni, F.: Dialectic proof procedures for assumption-based, admissible argumentation. Artificial Intelligence 170(2), 114–159 (2006)
6. Dung, P.M., Mancarella, P., Toni, F.: Computing ideal sceptical argumentation. Artificial Intelligence 171(10-15), 642–674 (2007)
7. van Eemeren, F.H., Grootendorst, R.: A Systematic Theory of Argumentation: The Pragma-Dialectical Approach. Cambridge University Press (2004)
8. Gordon, T.F.: An open, scalable and distributed platform for public discourse. In: Informatik. pp. 232–234 (2003)
9. Gürkan, A., Iandoli, L., Klein, M., Zollo, G.: Mediating debate through on-line large-scale argumentation: Evidence from the field. Information Sciences 180 (2010)
10. Karacapilidis, N.I., Papadias, D.: Computer supported argumentation and collaborative decision making: the HERMES system. Information Systems 26(4), 259–277 (2001)
11. Klein, M.: Enabling large-scale deliberation using attention-mediation metrics. Computer Supported Cooperative Work (CSCW) 21(4-5), 449–473 (2012)
12. Panzarino, O.: Learning Cypher. PACKT Publishing (2014)
13. Perelman, C.: The new rhetoric. In: Pragmatics of Natural Languages, pp. 145–149. Springer (1971)
14. Pollock, J.L.: Defeasible reasoning. Cognitive Science 11(4), 481–518 (1987)
15. Rahwan, I., Zablith, F., Reed, C.: Laying the foundations for a world wide argument web. Artificial Intelligence 171(10-15), 897–921 (2007)
16. Robinson, I., Webber, J., Eifrem, E.: Graph Databases. O'Reilly (2013)
17. Toulmin, S.E.: The Uses of Argument. Cambridge University Press (2003)
18. Walton, D., Reed, C., Macagno, F.: Argumentation Schemes. Cambridge University Press (2008)