Visual Gorgias: A Mechanism for the Visualization of an Argumentation Dialogue

ALEXANDROS VASSILIADES, School of Informatics, Aristotle University of Thessaloniki and Institute of Computer Science, Foundation for Research and Technology, Greece
ILIAS PAPADIMITRIOU, School of Informatics, Aristotle University of Thessaloniki, Greece
NICK BASSILIADES, School of Informatics, Aristotle University of Thessaloniki, Greece
THEODORE PATKOS, Institute of Computer Science, Foundation for Research and Technology, Greece

The area of eXplainable Artificial Intelligence (XAI) has been given great attention during the last decade, due to the obscurity that exists on how a data-driven model makes a decision. One idea in order to make Artificial Intelligence (AI) systems more explainable is to use cognitive methods that humans use when reasoning. For instance, argumentation is a common method that humans use when they explain an opinion [13]. In this paper, we introduce a framework that produces the visualization of an explanation from an argumentation dialogue, and can be easily understood by any user. More specifically, we utilize the argumentation tree that Gorgias returns [7] for the acceptability of an argument, to create a graph that represents the explanation of the argumentation tree.

CCS Concepts: • **General and reference** → *Experimentation*; • **Theory of computation** → *Semantics and reasoning*.

Additional Key Words and Phrases: Argumentation, Argumentation Dialogue, Explainability, Visualization

ACM Reference Format:
Alexandros Vassiliades, Ilias Papadimitriou, Nick Bassiliades, and Theodore Patkos. 2021. Visual Gorgias: A Mechanism for the Visualization of an Argumentation Dialogue. *J. ACM* 37, 4, Article 111 (August 2021), 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The area of eXplainable Artificial Intelligence (XAI) has been given great attention during the last decade, as data-driven models reached a level where they could solve decision-making problems or could make decisions for humans. For this reason, new regulations were defined by the EU [11] that made it mandatory for any AI system that makes a decision that may affect a human life, to have a mechanism that could explain to the user why it took the decision. Nevertheless, achieving an explainable system (i.e., a system that each one of its decision can be explained with a reasoning process) is a long-desired aspect for AI [9]. One idea is to use cognitive methods that we humans use in order to explain an opinion. A very common method of explaining our opinion is by using arguments, in order to show *how* and *why* we support (or decline) an opinion [13]. For this reason,

Authors' addresses: Alexandros Vassiliades, School of Informatics, Aristotle University of Thessaloniki, and Institute of Computer Science, Foundation for Research and Technology, Thessaloniki, Greece, valexande@csd.auth.gr; Ilias Papadimitriou, School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece, igpapadi@csd.auth.gr; Nick Bassiliades, School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece, nbassili@csd.auth.gr; Theodore Patkos, Institute of Computer Science, Foundation for Research and Technology, Heraklion, Greece, patkos@ics.forth.gr.

0004-5411/2021/8-ART111 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Argumentation Theory can provide reasoning that can explain why an opinion is acceptable or not, and by extension make AI systems more explainable.

An explanation from an AI system even when it is based on cognitive methods that are closer to the human way of explaining, can be obscure for people that are not related to the area of Mathematics and Computer Science. Therefore, our motivation in this paper is to provide a visualization method that translates the process of argumentative reasoning into a more clear and transparent explanation. Moreover, we would like to give ideas on how other XAI systems can translate their explanation, in order to become more user-friendly and gain the trust of their users.

The problem we are solving is the following: Given an argumentative reasoning process that supports (or attacks) an argument, we produce a small knowledge graph that explains in detail how the argumentation dialogue evolved in order for the argument to be accepted (or rejected). In our case, we use the argumentation tree [1] that is produced by the Prolog version of the Gorgias framework [7]. In our paper, one could find: (i) A user interface that receives an input in Gorgias format, and can be used in order to answer questions about facts, (ii) A visualization mechanism that translates the argumentation tree that Gorgias produces, into a graph, and (iii) A mechanism that associates the explanations of the attacks that the arguments receive with the nodes in the graph from (ii).

In the rest of the paper, we give the related work in Section 2. Next, in Section 3 initially, we describe the Gorgias framework which was developed with Prolog. Then, we show how we extended the Gorgias framework with a user interface, and the visualization mechanism that transforms the argumentation tree into a graph that explains the argumentation dialogue. Finally, in Section 4, we elaborate on how this mechanism can help other explainability mechanisms to become more user-friendly, and we provide a conclusion for the paper.

2 RELATED WORK

In the related work we elaborate over other frameworks that analyze an argumentation dialogue to make it more explainable for a user. In [14], the authors use a Decision Graph with Context (DGC), to understand the context, in order to support a decision. A DGC is a graph, where the decisions are represented as nodes and the interactions between them (attacking and supporting) as edges. The authors map the DGC in an assumption-based argumentation framework [12] by considering decisions as arguments and the interactions between them as attack and support relations. Then, if a decision is accepted in the assumption- based argumentation framework it is considered a good decision. There are differences with our study, as the authors in [14] give a theoretical framework in order to find the best decision while we provide a tool for visualizing why an argument is accepted or not. Moreover, our goal is to provide better explanations on why an argument holds while the authors of [14] find the best decision with their framework.

Two tools that use visualization to help empower the explanation capabilities of an AF can be found in [4, 6]. The first one is OpMap, a tool for visualizing large-scale opinions spaces into a geographical map. The goal of OpMap is to build a tool that can handle multi-opinion against or in favor of a topic, based on a SAF. OpMap first clusters the opinions using the GMap algorithm, which extends force-directed algorithms and constructs visualizations resembling a geographic map. After the clusters are created, OpMap maps the opinions-arguments and the clusters into a 2D map. The clusters are considered as columns and each argument as a row. Then, the arguments are given the label of True, False, or Judgement Suspension (i.e., Undefined). The second one is called AVIZE and is a tool for constructing arguments in the domain of international politics. AVIZE tries to create a triplet (topic; claim; premise) for each argument and it represents this as a table with a column for each component of the triplet. Obviously, AVIZE is a tool that helps to understand the structure of an argument and makes it easier for the user to find ways that can attack or support it.

Moreover, AVIZE is a supervised tool where the user must evaluate the quality of the triplet, and it can be used in other domains. The main difference between these studies with our study is that they analyze the structure of the argument. On other hand, we analyze the whole argumentation dialogue to provide a simpler explanation.

Interesting is the study of [15], in which the authors present a framework that is meant to help judges to define a sentence. Their framework compares current cases with past performed cases and returns the verdicts that were taken for similar cases. The authors in [8], with the use of argumentative semantics develop a mechanism that creates a visualization of the interactions between a set of rules, in defeasible logic. Also, the framework provides arguments and explanations to the judge, by showcasing the Redundant Attributes. Redundant Attributes are the parts in the description of a case that may contribute to the verdict, such as: the type of crime, amount of stolen items, condition of the accusant, crime evidence, number of abettors, and if the accusant is also the moral instigator. Of similar nature are the studies [2, 5]. The difference between the previous studies and our study is that the authors in all of them analyze the structure of the argument in order to give a better explanation. On the other hand, we analyze the whole argumentation dialogue to provide a simpler explanation.

3 VISUALIZATION OF EXPLANATION

In this section, we describe the Gorgias framework which we use as an underlying reasoning mechanism. Afterwards, we present how we visualize the argumentation dialogue by mapping it to a small knowledge graph that explains why an argument is acceptable or not.

3.1 The Gorgias Argumentation Framework

Argumentation technology caters for situations where systems need to support decision making under complex preference policies that consider diverse factors [3]. We can abstractly define argumentation as the mechanism that allows the interaction of different, possibly conflicting, arguments and provides the semantics for resolving conflicts. The conclusion supported by the arguments can be an action to take, the validity of a proposition, e.g. to validate or not a data access request.

A hierarchical argumentation framework uses object-level arguments for representing the decision policies and then priority arguments for expressing preferences on the object- level arguments to resolve possible conflicts. Additional priority arguments can be employed for resolving conflicts between priority arguments of a previous level. One such framework is Gorgias [7], which has been successfully applied in different applications.

Definition 3.1. A theory is a pair (T, P) whose sentences are formulae in the background monotonic logic $(L, \vdash)$ of the form $L \vdash L_1, \ldots, L_n$, where $L, L_1, \ldots, L_n$ are positive or negative ground literals. For rules in P the head L refers to an (irreflexive) higher priority relation, i.e. L has the general form $L = hp\ (rule_1, rule_2)$. The derivability relation, $\vdash$, of the background logic is given by the simple inference rule of modus ponens.

An *argument* for a literal L in a theory (T, P) is any subset, Arg, of this theory that derives $L, Arg \vdash L$, under the background logic. A part of the theory $T_0 \subseteq T$, is the *background theory* that contains facts and non-defeasible rules that always apply. An argument attacks another when they derive a contradictory conclusion. These two arguments are considered as *conflicting arguments*. A conflicting argument (from T) is admissible if it counter-attacks all the arguments that attack it. It counterattacks an argument if it can use priority arguments (from P) and make itself at least as strong as a conflicting argument.

Definition 3.2. An agent's argumentative policy theory is a theory $APT = ((T, T_0), PR, PC)$ where T contains the argument rules in the form of definite Horn logic rules, PR contains priority rules which are also definite Horn rules with head $hp(r_1, r_2)$ such that $r1, r2 \in T$ and all rules in PC are also priority rules with head $hp(R_1, R_2)$ such that $R_1, R_2 \in PR \cup PC$. T_0 contains auxiliary rules of the agent's background knowledge.

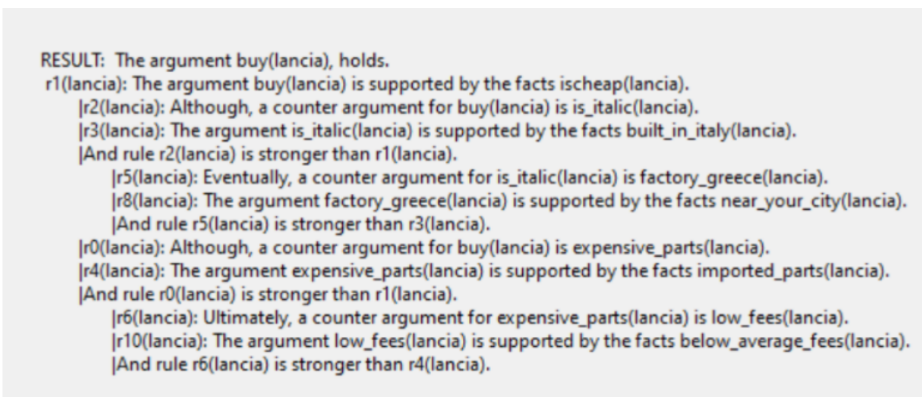
All in all, we specify rules in three different levels for defining a decision-making theory. The first level (or object-level) rules (T) refer directly to the subject domain and reflect the background knowledge needed for reaching the different goals. The second level rules define priorities over the first level rules, resolving possible conflicts. These situations usually reflect the needs of a role that the decision maker assumes or a context in which he finds himself, usually also including a default context. The third level (and also higher level) rules define priorities over the rules of the previous level but also over the rules of this level to define specific contexts, which can be specializations of the previous level contexts or their combinations.

Gorgias receives arguments in the format $rule(label\_argument(X), argument(X), [supporting\_facts(X)])$ where X is a fact. Moreover, facts have the format $rule(label\_fact, fact, [])$. Finally, preference rules have the format $rule(label\_preference(X), prefer(label\_argument_1(X), label\_argument_2(X)), [])$ where X is a fact, and $prefer(label\_argument_1(X), label\_argument_2(X))$ implies that Gorgias should consider the argument $label\_argument_1(X)$ stronger than $label\_argument_2(X)$, for the fact X.

Example 3.3 shows an argumentation theory that contains arguments against or in favor of the decision *to buy a car lancia.*

Example 3.3.
```
rule(r0(X), neg(buy(X)), [expensive_parts(X)]).
rule(r1(X), buy(X), [ischeap(X)]).
rule(r2(X), neg(buy(X)), [is_italian(X)]).
rule(r3(X), is_italian(X),[built_in_italy(X)]).
rule(r4(X), expensive_parts(X),[imported_parts(X)]).
rule(r6(X), neg(expensive_parts(X)), [low_fees(X)]).
rule(r5(X), neg(is_italian(X)), [factory_greece(X)]).
rule(r7(X), neg(is_italian(X)), [high_refund(X)]).
rule(r8(X), factory_greece(X), [near_your_city(X)]).
rule(r9(X), high_refund(X), [high_resale(X)]).
rule(r10(X), low_fees(X), [below_average_fees(X)]).
%% Facts
rule(myfact1, ischeap(lancia), []).
rule(myfact2, built_in_italy(lancia), []).
rule(myfact3, imported_parts(lancia), []).
rule(myfact4, near_your_city(lancia), []).
rule(myfact5, below_average_fees(lancia), []).
rule(myfact6, high_resale(lancia), []).
%% Preference rules
rule(pr0(X), prefer(r0(X),r1(X)), []).
rule(pr1(X), prefer(r2(X),r1(X)), []).
rule(pr2(X), prefer(r5(X),r3(X)), []).
rule(pr3(X), prefer(r6(X),r4(X)), []).
rule(pr4(X), prefer(r7(X),r3(X)), []).
```

Given the Example 3.3, Gorgias will return the argumentation dialogue as shown in Figure 1. Gorgias, answer to the question *"Should I buy a car lancia?"* is positive (i.e., the argument *buy(lancia)* is acceptable), because even though the argument *buy(lancia)* receives two attacks from the arguments *is_italian(lancia)*, *expensive_parts(lancia)* which are supported by the facts *built_in_italy(lancia)* and *imported_parts(lancia)*, respectively. The argument *buy(lancia)* is defended by the argument *factory_greece(lancia)* which attacks the argument is_italian(lancia) that is supported by the fact *near_your_city(lancia)*, and the argument *low_fees(lancia)* which attacks the argument expensive_parts(lancia) that is supported by the fact *below_average_fees(lancia)*.



RESULT: The argument buy(lancia), holds.
r1(lancia): The argument buy(lancia) is supported by the facts ischeap(lancia).
|r2(lancia): Although, a counter argument for buy(lancia) is is_italic(lancia).
|r3(lancia): The argument is_italic(lancia) is supported by the facts built_in_italy(lancia).
|And rule r2(lancia) is stronger than r1(lancia).
|r5(lancia): Eventually, a counter argument for is_italic(lancia) is factory_greece(lancia).
|r8(lancia): The argument factory_greece(lancia) is supported by the facts near_your_city(lancia).
|And rule r5(lancia) is stronger than r3(lancia).
|r0(lancia): Although, a counter argument for buy(lancia) is expensive_parts(lancia).
|r4(lancia): The argument expensive_parts(lancia) is supported by the facts imported_parts(lancia).
|And rule r0(lancia) is stronger than r1(lancia).
|r6(lancia): Ultimately, a counter argument for expensive_parts(lancia) is low_fees(lancia).
|r10(lancia): The argument low_fees(lancia) is supported by the facts below_average_fees(lancia).
|And rule r6(lancia) is stronger than r4(lancia).

Fig. 1. Gorgias output

3.2 Visual Gorgias

The visualization mechanism that we have developed uses Prolog files as in Example 3.3, in order to produce the explanation of the argumentation dialogue. The user has to provide the Prolog file and the question that she wishes to address to Gorgias. For instance, Figure 2 shows the initial window of our user interface where the user gives the path of the Prolog file, and the argument that she wants to validate. Based on Example 3.3 the user wants to verify the acceptability of the argument that it is worthy to buy a lancia car.

After hitting the Run button, the Gorgias framework will respond to our query, and a new window will be created. This window shows the visualization of the Gorgias response in an argumentation tree. In this graph, each node represents an argument and the arrow from a node to another represents that the first argument attacks the second one, namely, it as a counterargument for the second argument. The root node is the argument we used in our query and its color is green in case the argument holds and red in case the argument does not hold. Based on the annotation from Figure 1 (for the arguments) we see in Figure 3 the knowledge graph, $r1$ is the node our query was about, $r2$ and $r0$ are counterarguments against $r1$, whereas $r5$ and $r6$ are counterarguments against $r2$ and $r0$, respectively. Since there is no counterargument against $r5$ or $r0$, the argument buy(lancia) that the question was about holds, and therefore the root node is green. Notice, that if we consider that the root of the tree has depth 1, then every layer at even depth defends the root, and every layer at odd depth attacks the root.

In Figure 3, each node represents an argument. Each argument is supported by facts and there is a preference rule, which proves that this argument is stronger than the one that it attacks. The user can click on a node and a new window will pop-up, showing all available information about this

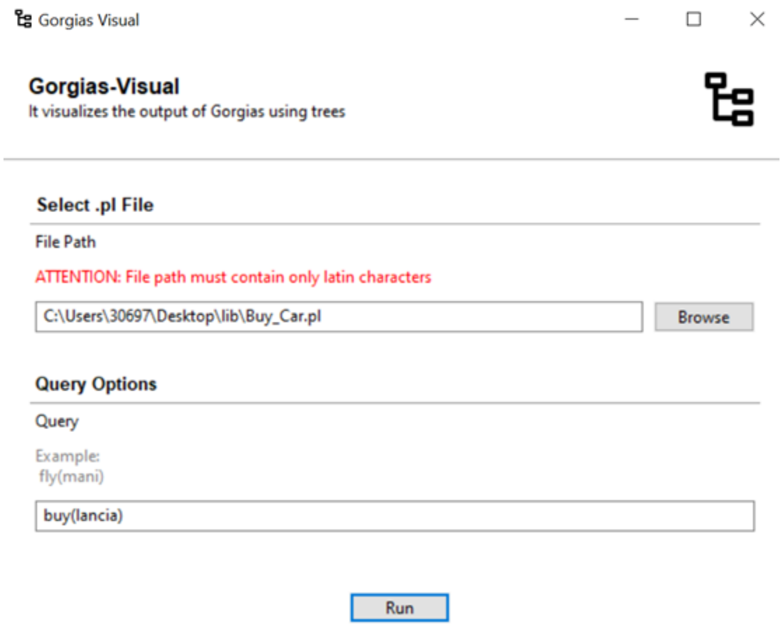


Fig. 2. Input of Visualization Mechanism

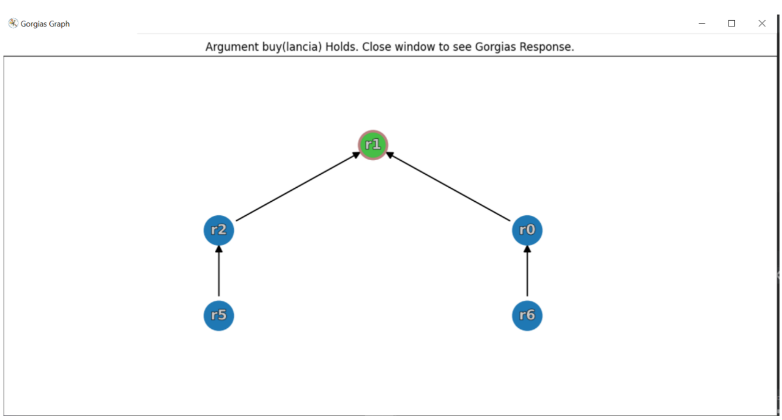


Fig. 3. Visualization of the Argumentation Dialogue from Gorgias

node. The text displayed in this window will show which argument is attacked by this argument, which preference rule shows that it is stronger than the argument it attacks, and which facts support this argument.

For example, in case we click at the node $r2$ in Figure 3, a window will appear, where all details about this node will be displayed (Figure 4). The first line explains that $r1$ is attacked by $r2$, the second line shows the facts which support $r2$, and the third line describes the preference rule that makes $r2$ stronger than $r1$.The visualization mechanism was developed with *Python* using the *pyswip*[1] and the *pyplot*[2] modules, while the underlying reasoning mechanism Gorgias was developed with *SWI-Prolog*[3].

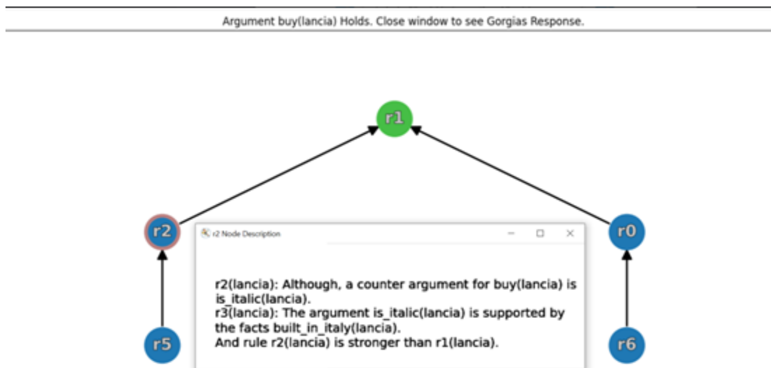


Fig. 4. Visualization of the Argumentation Dialogue from Gorgias with Textual Description

4 DISCUSSION AND CONCLUSION

In this paper, we introduced a visualization process of an argumentation dialogue in order for the dialogue to become easily explainable by a system. Our main contribution is that we propose a new method of providing explanations through graphical representations, which will help a user not related to Computer Science, understand why the system took a decision. The method can become more generic, as instead of the argumentation dialogue which is produced by Gorgias, the method can be fine-tuned in order to reason over a set of rules in non-monotonic or defeasible logic.

Moreover, this method of graphical representation of explanations can work as a starting point to other AI systems that can provide explanations. As it is commonly known, many AI systems, due to the fact that they are oriented for programmers or researchers, many times they provide explanations in formats that are obscure for a user that is not related to Computer Science. Therefore, through this paper we want to motivate others to follow a similar direction and make the explanations of their AI system more transparent. Similarly to what the authors in [10] did, to explain data-driven models decisions using argumentation.

As noticed we omitted the evaluation process mainly for two reasons. Firstly, the rationality and evaluation of the underlying argumentation system Gorgias was already performed by studies before us [1, 7]. As Gorgias is one of the off-the-shelf tools for reasoning with non-monotonic and defeasible logic. Secondly, the evaluation of the graphical representation of the explanation required a user evaluation in order to estimate how satisfied they are with the representation. But due to the fact that we have not yet developed a web API for our framework, we let the user evaluation as a future work.

Thus, in the future we intend to develop a web API for our framework in order for any user to have access to it. Furthermore we will perform a user evaluation that will showcase the quality of the graphs that our framework produces, with respect to how understandable they are. Finally, we intend to explore alternative underlying reasoning mechanisms on defeasible and non-monotonic logics, in addition to Gorgias.

[1] https://pypi.org/project/pyswip/
[2] https://matplotlib.org/stable/tutorials/introductory/pyplot.html
[3] https://www.swi-prolog.org

5 ACKNOWLEDGMENTS

This project has received funding from the Hellenic Foundation for Research and Innovation (HFRI) and the General Secretariat for Research and Technology (GSRT), under grant agreement No 188. We thank Prof. Antonis Kakas (University of Cyprus) for the provision of Gorgias and for fruitful discussions on argumentation.

REFERENCES

[1] Nick Bassiliades, Nikolaos I Spanoudakis, and Antonis C Kakas. 2018. Towards multipolicy argumentation. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence*. ACM, 1–10.

[2] Trevor JM Bench-Capon. 2020. Before and after Dung: Argumentation in AI and Law. *Argument & Computation* 11, 1-2 (2020), 221–238.

[3] Trevor JM Bench-Capon and Paul E Dunne. 2007. Argumentation in artificial intelligence. *Artificial intelligence* 171, 10-15 (2007), 619–641.

[4] Gregor Betz, Michael Hamann, Tamara Mchedlidze, and Sophie von Schmettow. 2019. Applying argumentation to structure and visualize multi-dimensional opinion spaces. *Argument & Computation* 10, 1 (2019), 23–40.

[5] Martin Caminada, Samy Sá, Joao Alcântara, and Wolfgang Dvořák. 2015. On the equivalence between logic programming semantics and argumentation semantics. *International Journal of Approximate Reasoning* 58 (2015), 87–111.

[6] Nancy L Green, Michael Branon, and Luke Roosje. 2019. Argument schemes and visualization software for critical thinking about international politics. *Argument & Computation* 10, 1 (2019), 41–53.

[7] Antonis C Kakas, Pavlos Moraitis, and Nikolaos I Spanoudakis. 2019. GORGIAS: Applying argumentation. *Argument & Computation* 10, 1 (2019), 55–81.

[8] Efstratios Kontopoulos, Nick Bassiliades, and Grigoris Antoniou. 2011. Visualizing Semantic Web proofs of defeasible logic in the DR-DEVICE system. *Knowledge-Based Systems* 24, 3 (2011), 406–419.

[9] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence* 267 (2019), 1–38.

[10] Ioannis Mollas, Nick Bassiliades, and Grigorios Tsoumakas. 2020. Altruist: Argumentative Explanations through Local Interpretations of Predictive Models. arXiv:2010.07650 [cs.LG]

[11] General Data Protection Regulation. 2016. Regulation EU 2016/679 of the European Parliament and of the Council of 27 April 2016. *Official Journal of the European Union* (2016).

[12] Francesca Toni. 2014. A tutorial on assumption-based argumentation. *Argument & Computation* 5, 1 (2014), 89–117.

[13] Alexandros Vassiliades, Nick Bassiliades, and Theodore Patkos. 2021. Argumentation and explainable artificial intelligence: a survey. *The Knowledge Engineering Review* 36 (2021).

[14] Zhiwei Zeng, Chunyan Miao, Cyril Leung, and Jing Jih Chin. 2018. Building more explainable artificial intelligence with argumentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[15] Qiaoting Zhong, Xiuyi Fan, Francesca Toni, and Xudong Luo. 2014. Explaining Best Decisions via Argumentation.. In *ECSI*. 224–237.