José Manuel Perdigão Venâncio

BSc in Computer Science

A Tool for Online Debates

Dissertation plan submitted in partial fulfillment
of the requirements for the degree of

Master of Science in
Computer Science and Engineering

Adviser: João Leite, Associate Professor,
Faculdade de Ciências e Tecnologia,
NOVA University of Lisbon

Co-advisers: Teresa Romão, Assistant Professor,
Faculdade de Ciências e Tecnologia,
NOVA University of Lisbon

João Leitão, Assistant Professor,
Faculdade de Ciências e Tecnologia,
NOVA University of Lisbon

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

February, 2017

Abstract

Nowadays, internet is one of the most used vehicles to connect people. There are some social networks, such as Facebook, Twitter, and Youtube that provide ways for users to interact among them, via comment boxes. However, these interactions usually leads to unstructured or even chaotic debates. This problem was first described by Leite and Martins in [24]. In their work, semantics for Social Abstract Argumentation were also proposed in order to tackle this problem. Although these semantics have been proposed, there is no tool that can validate this model.

Our aim is to develop an online debate tool based on Social Abstract Argumentation, in order to validate this model. To achieve this, we will adapt an algorithm that implements Social Abstract Argumentation efficiently [15]. The system that we aim to develop, must rely on a robust architecture, since it will allow a large number of users, arguments, and relations, while providing an adequate performance when running the algorithm. This tool must be easy to use as well as provide a pleasant user experience. Furthermore, it can be used in a vast range of applications (e.g. online newspapers, participatory democracy, decision making) with a significant impact.

Keywords: Online Debate, Abstract Argumentation Framework, Ranking-based Semantics for Abstract Argumentation, Social Abstract Argumentation

Resumo

A internet é uma das formas mais usadas para ligar pessoas. Existem algumas redes sociais como o Facebook, Twitter e Youtube que fornecem meios de interação entre utilizadores, através de caixas de comentários. O resultado deste tipo de interações leva a debates desestruturados ou até mesmo caóticos. Este problema foi primeiramente descrito por Leite e Martins em [24]. No seu trabalho foram também propostas semânticas para Social Abstract Argumentation, com o intuito de resolver este problema. Apesar de estas semânticas terem sido propostas, não existe qualquer ferramenta que possa validar este modelo.

O nosso objetivo é desenvolver uma ferramenta de debate online baseada na Social Abstract Argumentation, de forma a validar este modelo. Para o atingir, iremos adaptar um algoritmo que implementa a Social Abstract Argumentation de forma eficiente [15]. O sistema que pretendemos desenvolver deve ter por base uma arquitetura robusta, visto que irá permitir um elevado número de utilizadores, argumentos e relações, possibilitando uma performance adequada aquando da execução do algoritmo. Esta ferramenta deve ser fácil de usar e proporcionar uma experiência agradável ao utilizador. Para além disto, esta ferramenta poderá ser utilizada num vasto leque de aplicações (e.g. jornais online, democracia participativa, tomada de decisão) com um grande impacto.

Palavras-chave: Debate Online, Framework de Argumentação Abstracta, Semânticas baseadas em ranking para Argumentação Abstracta, Argumentação Abstracta Social

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Context

The word *debate* is often used to define a discussion between two or more individuals. Every time two or more people gather around they tend to start a debate. However, in some countries, such as the United States of America, debate can be considered to be a sport and, as any other sports, it has its own rules. Depending on these rules, the debate can be classified in many different categories, from Judicial Debate or Parliamentary Debate ruled by the court of law and the parliamentary procedure, respectively, to Academic Debate that usually has its own rules depending on the institution or contest where it takes place [23]. All of these categories of debates have a common denominator, they all have intrinsic rules for the debate. Nonformal Debate is the only form of debate that is conducted without formal rules found in the categories mentioned before [23]. Although this category does not rely on formal rules, it is probably the most used form of debate. This is the kind of debate that one can have in a bar with their friends when arguing about the best beer in the table.

Since this form of debate is so simple, it also emerges naturally in society. Nonformal debate occurs in every kind of media platforms, such as television and radio, and also over the Web. Most of the newspapers websites provide a comment area associated to each article, where the users can express their opinion about the subject. The same model can be seen in social networks (e.g Facebook, Twitter and Youtube) where a user can freely comment on every post of the network. Despite of the freedom given to the users, these kinds of models usually lead to debates that easily deviate from the main topic, considering that not all of the users are serious and some of them just want to express their emotions, despite the main topic or main arguments. Additionally, in the platforms mentioned before, the way the debate is presented is often based on a timeline, where on top the most recent comments appear.

The problem of unstructured or even chaotic debates in Social Web was first described in [24]. In their work, semantics for Social Abstract Argumentation were also proposed in order to tackle this problem. These semantics are one of the first extensions, for the Dung's Abstract Argumentation Framework, which proposed its application to Social Networks. Dung formally defined what is an Abstract Argumentation Framework [19] and also defined Abstract Argumentation. Dung defined that in Abstract Argumentation, an argument is an abstract entity whose role is solely determined by its relations to the other arguments, the content of the argument itself is not relevant. Among other things, he defined attacks between arguments, the acceptance of an argument and also some properties of an Abstract Argumentation Framework. The extension proposed by Leite and Martins [24] introduced voting as a new factor that would have impact on the score of each argument. Social Abstract Argumentation was after extended to allow votes on attacks [21], hence providing the user a new tool that would have impact on the argument score. Although these semantics have been proposed, there is no tool (that we are aware of) that can validate this model. Therefore, we want to develop a platform that will allow the users to freely debate, however providing relevant assistance to follow and understand the major points of these debates. To achieve these goals, our tool will be based on Social Abstract Argumentation. An efficient algorithm to the Social Abstract Argumentation has also been proposed and implemented [15]. Nevertheless, some different semantics for Abstract Argumentation were proposed since this one, that may be also relevant to achieve our goals.

The graph in figure 1.1 illustrates an example of a Social Abstract Argumentation Framework. In this graph, each node represents an argument and each edge represents attacks between arguments, both of them have the values of positive and negative votes associated to them.

The main topic of the debate illustrated in figure 1.1, is the benefits of using e-cigarettes over regular cigarettes [1]. Each node stands for an argument and has its respective votes, each edge stands for an attack relation and also has its respective votes. For example, the argument f attacks argument a, since the argument a states that the use of e-cigarettes can be safer than the use of regular cigarettes, and the argument f points out a study about people that got seriously hurt with the batteries explosions of e-cigarettes. Let us consider the example in figure 1.1, in order to better understand the votes on attacks [21]. Argument h has 54 positive votes and 22 negative votes. Also h attacks c, however as that attack has 48 negative votes and only 23 positive votes, the impact of that attack is reduced.

We aim at designing a platform that lets the user freely participate, using any kind of content or simply participate because he agrees with a few arguments from each side. Additionally, the user may as well have a really different position about the discussed

[1]The content of the arguments were retrieved and adapted from the ARG-tech Argument Mining datasets [4]

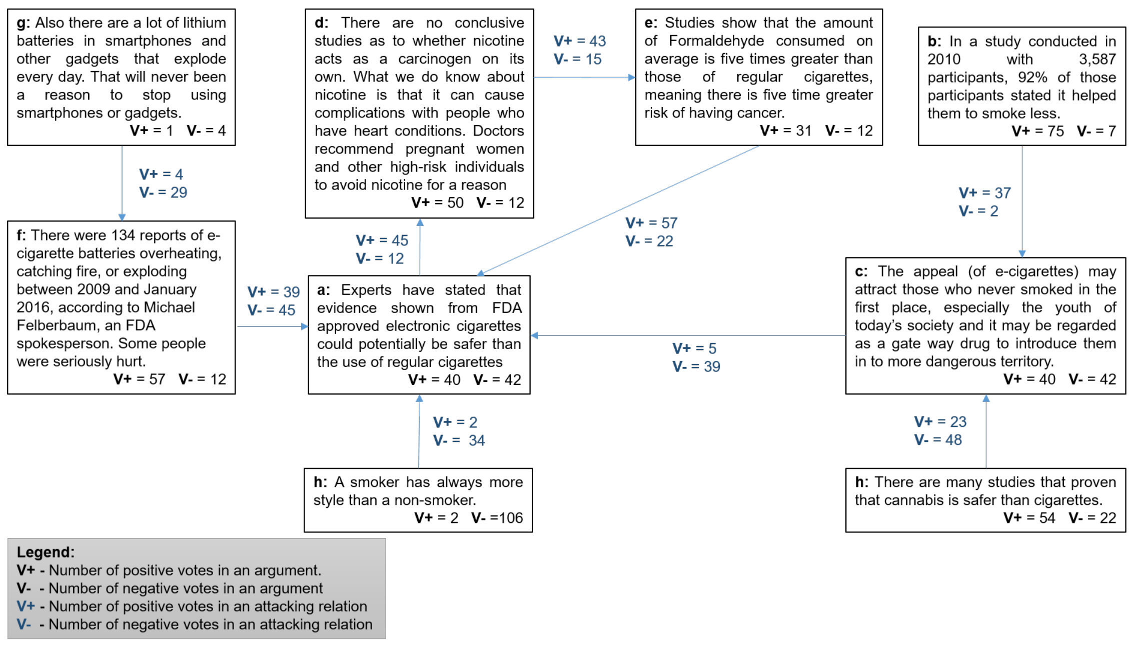

g: Also there are a lot of lithium batteries in smartphones and other gadgets that explode every day. That will never been a reason to stop using smartphones or gadgets. V+ = 1 V- = 4
d: There are no conclusive studies as to whether nicotine acts as a carcinogen on its own. What we do know about nicotine is that it can cause complications with people who have heart conditions. Doctors recommend pregnant women and other high-risk individuals to avoid nicotine for a reason V+ = 50 V- = 12
V+ = 43 V- = 15
e: Studies show that the amount of Formaldehyde consumed on average is five times greater than those of regular cigarettes, meaning there is five time greater risk of having cancer. V+ = 31 V- = 12
b: In a study conducted in 2010 with 3,587 participants, 92% of those participants stated it helped them to smoke less. V+ = 75 V- = 7
V+ = 4 V- = 29
V+ = 37 V- = 2
V+ = 57 V- = 22
V+ = 45 V- = 12
f: There were 134 reports of e-cigarette batteries overheating, catching fire, or exploding between 2009 and January 2016, according to Michael Felberbaum, an FDA spokesperson. Some people were seriously hurt. V+ = 57 V- = 12
V+ = 39 V- = 45
a: Experts have stated that evidence shown from FDA approved electronic cigarettes could potentially be safer than the use of regular cigarettes V+ = 40 V- = 42
V+ = 5 V- = 39
c: The appeal (of e-cigarettes) may attract those who never smoked in the first place, especially the youth of today's society and it may be regarded as a gate way drug to introduce them in to more dangerous territory. V+ = 40 V- = 42
V+ = 2 V- = 34
V+ = 23 V- = 48
h: A smoker has always more style than a non-smoker. V+ = 2 V- =106
h: There are many studies that proven that cannabis is safer than cigarettes. V+ = 54 V- = 22
Legend:
V+ - Number of positive votes in an argument.
V- - Number of negative votes in an argument
V+ - Number of positive votes in an attacking relation
V- - Number of negative votes in an attacking relation

Figure 1.1: Social Abstract Argumentation Framework example

thesis, thus supporting a new side. The system should have a proper feedback, showing the user the most relevant arguments, in order to produce a pertinent outcome to the debate. To achieve this, every argument should have its own score and all the users should be allowed to express their opinion in different ways. The platform we aim to design should promote wide scale richer interactions in the form of debate, therefore it must facilitate:

- More open and detailed participation [24]. The user should be able to participate in different ways (e.g. add a new argument, add an attack relation between arguments, vote on an argument, and vote on an attack relation), without needing to know the formal rules of the debate.

- More flexible participation [24]. The tool must not be confined to only two sides in a debate. There may be more than two sides, and the user should be allowed to propose arguments for more than one side of the debate.

- Appropriate feedback to users [24]. It should be easy to assess the strength of each argument, taking into account the interactions performed by the users of the framework.

- The content of an argument can be a content of any type: text, video, link, image, document.

There are already some systems on the internet that allow the users to debate. Some of these systems were designed to a more serious debate, to ground a thesis, or to share different points of view and ideas, createdebate.com and agora.gatech.edu are some of the examples that we can find over the web. In some of these platforms the debate may have some formal rules that are enforced, and mechanisms to restrain the user participation in the debate (e.g. the user may participate but has to formally define every relation between all the parts of the argument). There are also other debate tools, that are less formal, where an user can add new arguments to one or both sides of the thesis being debated, as well as support or dispute other arguments. Both categories have specific objectives: in one hand, if we consider less formal debate tools, they encourage the users to share points of view in a more relaxed way, so it is not essential to avoid losing the main topic. On the other hand, the platforms that rely more on formal rules, seem to be perfect to ground a thesis, albeit being restrictive to users. Despite their merits, these tools have several characteristics that may limit their adoption in a wide Social Web, namely [21]:

1. In some of these tools, only two antagonistic users can engage in a debate, others can vote for the winning side, but not on concrete arguments.

2. The debate structure is sometimes very rigid, with a pre-fixed number of rounds and strict debate rules not known by most users.

3. Only a few of these tools provide facilities to reuse arguments (e.g. `consider.it` [14]) and debates.

4. Most of the times the debate stops short of reasoning with the debate data and votes/opinions, yielding to simplistic and naïve outcomes.

Although these tools do not serve our purpose, we can draw inspiration on some solutions of these systems when designing our tool.

We want to provide a tool that allows a large number of users to debate a topic, furthermore each user can be participating in more than one debate. To achieve this, the system must rely on a robust architecture, allowing a large number of users, arguments, and relations, while providing an adequate performance when updating the scores associated with each argument. As this is not a trivial task to do, we are going to explore some different alternatives to approach this problem in section 2.3. Moreover, the tool that we are going to develop should be possible to be used by any kind of user, so it also must be easy to use as well as provide a pleasant user experience. Furthermore, this tool could be used in a vast range of applications (e.g. online newspapers, participatory democracy, decision making) with a significant impact.

1.2 Objectives

The main goal of the work discussed in this document is to develop a web-based plug-in that will provide the user all the features and properties mentioned in the previous section.

In order to, develop this kind of application we must certify that it follows some unavoidable principles in a distributed system:

Scalability. The system must be able to deal with an increasing number of users and arguments and maintain its performance.

Security. The system must be able to keep the integrity and confidentiality of the data.

Performance. The system should run the valuation algorithms in a small amount of time, therefore both the algorithms and the system have to be optimized.

Also we need to provide the user a system that is both simple and complete:

Learnability. The user should be able to learn how to use the system fast. [27]

Efficiency. An experienced user should be able to achieve a high level of productivity. An user should be able to complete a task in the appropriate period of time, and without much effort.[27]

Memorability. The system should be easy to remember after a short or long period without using it.[27]

Satisfaction. The system should be pleasant to use.[27]

We propose a web plugin, that can be easily embedded in a website, which will allow the users to freely debate. This tool will be based on Social Abstract Argumentation [24] and its respective algorithm [15].

Our prototype will mainly consist in two different parts. The first part will be a distributed-system, that will provide all the necessary storage space and computational power, where the algorithm and all the logic will be implemented. The second part will consist in a web server that will provide the web plugin in a way that it can be embedded in a website.

1.3 Contributions and Document Structure

Abstract Argumentation is one of the core studies in Artificial Intelligence. Since 1995 a lot of work has been done in this field of study. However there are a lot of proposed semantics and algorithms, there are almost no prototypes of working tools in this field of study.

With the work to be conducted in the context of the MSc dissertation, we aim to build and evaluate a prototype that can serve as a proof of concept for Social Abstract Argumentation and also a tool to showcase this platform and test its acceptance among the users.

The remainder of this document is structured as follows:

Chapter 2 - State of the Art. This chapter is divided into three different sections. In the first section we will discuss the different models explored by the existing applications. In the second section we will describe the existing semantics to approach our problem. Finally, in the last section, we will study different programming models that may be employed to build our platform.

Chapter 3 - Methodology. In this chapter we describe the tools and technologies that we are going to use as well as the system architecture. We will also define the metrics that are going to be used in order to evaluate the resulting system. Lastly, a project schedule is presented along with a summary.

Chapter 2

State of the Art

In this chapter we will start by discussing the existing categories of applications that may address our problem. Afterwards, we will describe the existing ranking-based semantics for Abstract Argumentation. Finally, we will see three different architectures that can be used for building a prototype of the framework and discuss the best option in light of our objectives.

2.1 Existing argumentation/debate applications

There are already a lot of debate systems in the web that we will analyse throughout this section. In order to do that, we will divide the different systems in classes and discuss the different features for each one.

In this section we will consider two main types of existing applications. First, we will see the systems that are based on Abstract Argumentation, although we will use a broader definition of Abstract Argumentation, where the only property is that an argument is not divided into parts. Afterwards we will study the systems based on structured argumentation[1].

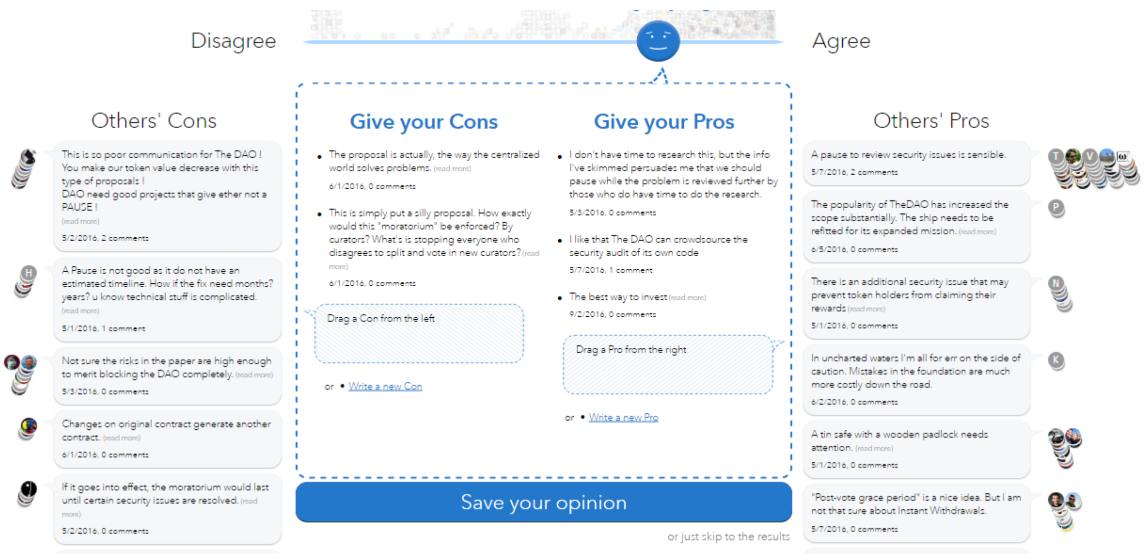
2.1.1 Abstract Argumentation Systems

In this section we are going to present systems that allow users to have a debate based on abstract argumentation.

[1] An argument is said to be structured if the premises and claim are explicit, and the relationship between them is formally defined[7].

2.1.1.1 Pros and Cons

In this class of solutions, only one thesis is presented to the user and then users are given the possibility to discuss the pros and the cons of this thesis. In most models, the user can contribute to both sides of the thesis. The systems `consider.it` [14] and `createdebate.com` [28] are two good examples of this kind of systems. Consider.it lets the user add an argument for or against the main thesis. It also lets the user express his/her opinion by asking how much does he/she supports one or other side of the thesis. An user can also reuse existing arguments to express his/her point of view (similar to vote on an argument, see figure 2.1).


Disagree

Agree

Others' Cons

This is so poor communication for The DAO ! You make our token value decrease with this type of proposals !
DAO need good projects that give ether not a PAUSE !
(read more)
5/2/2016, 2 comments

A Pause is not good as it do not have an estimated timeline. How if the fix need months? years? u know technical stuff is complicated.
(read more)
5/1/2016, 1 comment

Not sure the risks in the paper are high enough to merit blocking the DAO completely. (read more)
5/3/2016, 0 comments

Changes on original contract generate another contract. (read more)
6/1/2016, 0 comments

If it goes into effect, the moratorium would last until certain security issues are resolved. (read more)
5/2/2016, 0 comments

Give your Cons

- The proposal is actually, the way the centralized world solves problems. (read more)
  6/1/2016, 0 comments
- This is simply put a silly proposal. How exactly would this "moratorium" be enforced? By curators? What's is stopping everyone who disagrees to split and vote in new curators? (read more)
  6/1/2016, 0 comments

Drag a Con from the left

or • Write a new Con

Give your Pros

- I don't have time to research this, but the info I've skimmed persuades me that we should pause while the problem is reviewed further by those who do have time to do the research.
  5/3/2016, 0 comments
- I like that The DAO can crowdsource the security audit of its own code
  5/7/2016, 1 comment
- The best way to invest (read more)
  9/2/2016, 0 comments

Drag a Pro from the right

or • Write a new Pro

Save your opinion

or just skip to the results

Others' Pros

A pause to review security issues is sensible.
5/7/2016, 2 comments

The popularity of TheDAO has increased the scope substantially. The ship needs to be refitted for its expanded mission. (read more)
6/5/2016, 0 comments

There is an additional security issue that may prevent token holders from claiming their rewards (read more)
5/1/2016, 0 comments

In uncharted waters I'm all for err on the side of caution. Mistakes in the foundation are much more costly down the road.
6/2/2016, 0 comments

A tin safe with a wooden padlock needs attention. (read more)
5/1/2016, 0 comments

"Post-vote grace period" is a nice idea. But I am not that sure about Instant Withdrawals.
5/7/2016, 0 comments

Figure 2.1: Consider.it

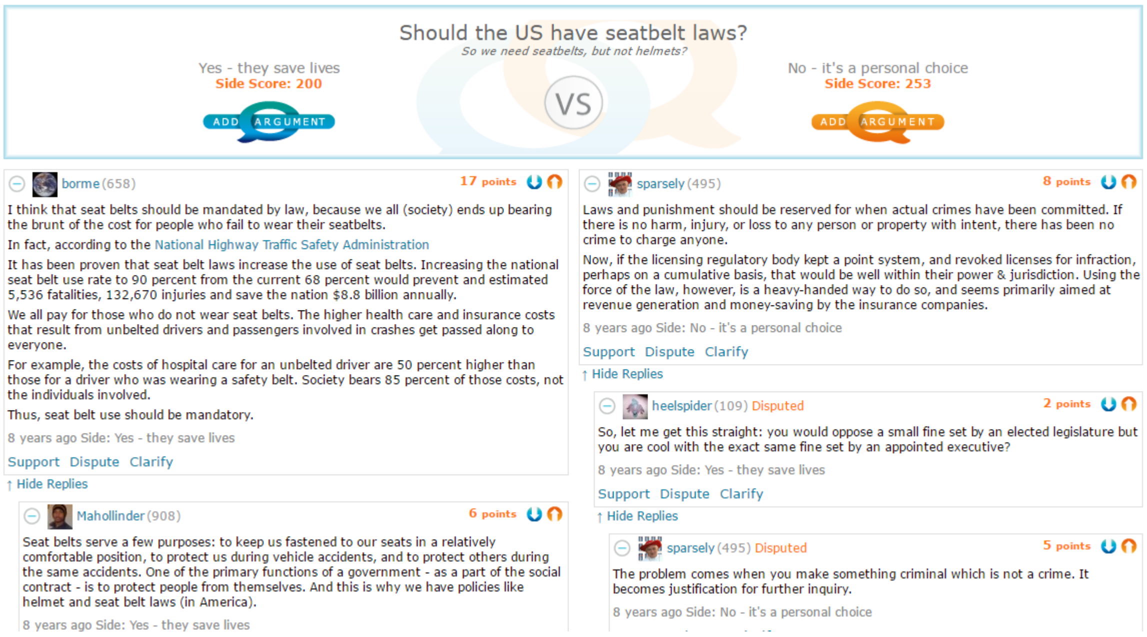
CreateDebate also lets the user add arguments to both sides of the thesis, although it differs from the previous system because the user can dispute, clarify, or support any argument as well as upvote or downvote each one of them (see figure 2.2).

Clearly the main purpose of this type of applications is to freely discuss a single thesis, without having to be concerned with whether or not some argument is valid, since the judgement is left to the users of the system.

2.1.1.2 Mind Map Systems

Another way to organize arguments is to use a mind map, these applications are not argumentation systems. In this kind of applications the user starts with a main idea and decompose the idea into smaller ones. In this section, we are going to examine two different systems that are based in this concept, `debategraph.org`[17] and `mindmeister.com`[26].

Debategraph starts with a main idea in the center node of the graph and the subideas all around the center node connected to that main idea. Everytime an user clicks on a node, that node goes to the center and the other nodes disappear, appearing new nodes



Should the US have seatbelt laws?

So we need seatbelts, but not helmets?

Yes - they save lives
Side Score: 200

No - it's a personal choice
Side Score: 253

borme (658) 17 points

I think that seat belts should be mandated by law, because we all (society) ends up bearing the brunt of the cost for people who fail to wear their seatbelts.

In fact, according to the National Highway Traffic Safety Administration

It has been proven that seat belt laws increase the use of seat belts. Increasing the national seat belt use rate to 90 percent from the current 68 percent would prevent and estimated 5,536 fatalities, 132,670 injuries and save the nation $8.8 billion annually.

We all pay for those who do not wear seat belts. The higher health care and insurance costs that result from unbelted drivers and passengers involved in crashes get passed along to everyone.

For example, the costs of hospital care for an unbelted driver are 50 percent higher than those for a driver who was wearing a safety belt. Society bears 85 percent of those costs, not the individuals involved.

Thus, seat belt use should be mandatory.

8 years ago Side: Yes - they save lives

Support Dispute Clarify

↑ Hide Replies

Mahollinder (908) 6 points

Seat belts serve a few purposes: to keep us fastened to our seats in a relatively comfortable position, to protect us during vehicle accidents, and to protect others during the same accidents. One of the primary functions of a government - as a part of the social contract - is to protect people from themselves. And this is why we have policies like helmet and seat belt laws (in America).

8 years ago Side: Yes - they save lives

sparsely (495) 8 points

Laws and punishment should be reserved for when actual crimes have been committed. If there is no harm, injury, or loss to any person or property with intent, there has been no crime to charge anyone.

Now, if the licensing regulatory body kept a point system, and revoked licenses for infraction, perhaps on a cumulative basis, that would be well within their power & jurisdiction. Using the force of the law, however, is a heavy-handed way to do so, and seems primarily aimed at revenue generation and money-saving by the insurance companies.

8 years ago Side: No - it's a personal choice

Support Dispute Clarify

↑ Hide Replies

heelspider (109) Disputed 2 points

So, let me get this straight: you would oppose a small fine set by an elected legislature but you are cool with the exact same fine set by an appointed executive?

8 years ago Side: Yes - they save lives

Support Dispute Clarify

↑ Hide Replies

sparsely (495) Disputed 5 points

The problem comes when you make something criminal which is not a crime. It becomes justification for further inquiry.

8 years ago Side: No - it's a personal choice

Figure 2.2: CreateDebate

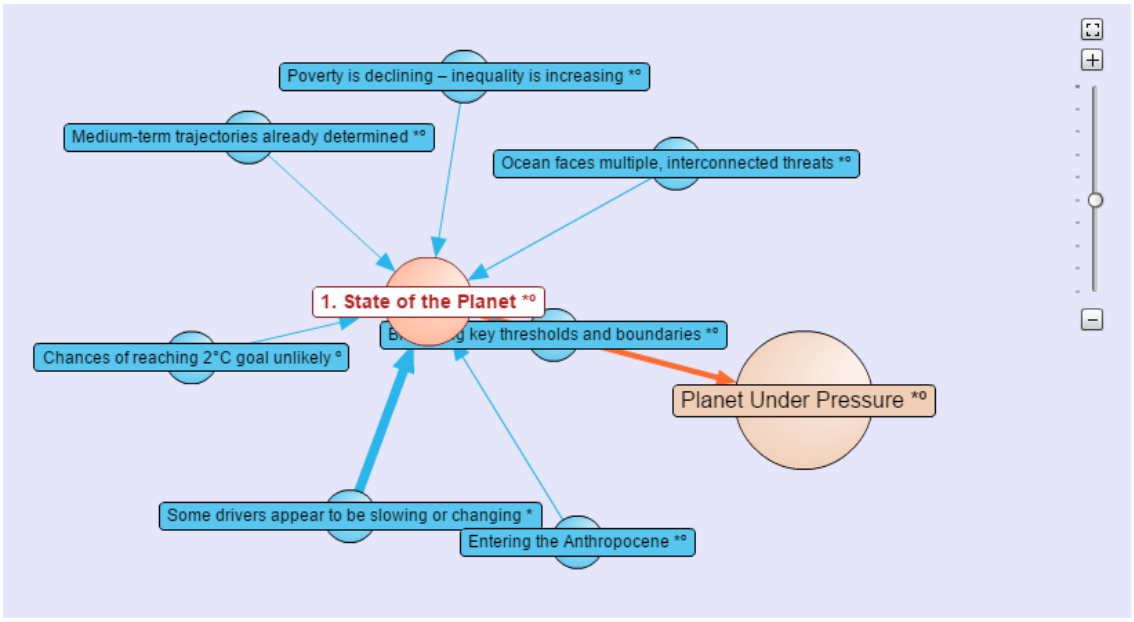

Poverty is declining – inequality is increasing *°
Medium-term trajectories already determined *°
Ocean faces multiple, interconnected threats *°
1. State of the Planet *°
Br...g key thresholds and boundaries *°
Chances of reaching 2°C goal unlikely °
Planet Under Pressure *°
Some drivers appear to be slowing or changing *
Entering the Anthropocene *°

Figure 2.3: Debategraph interface

all around the center node, representing the subideas of the current selected node (see figure 2.3).

MindMeister is a mind map. In the center we get the main idea and as we get further from the center we start to get that idea decomposed into smaller branches (see figure 2.4).

These kind of systems are used to expose and navigate all the smaller aspects of a main thesis, this is useful to organize a discussion or even to study some subject without

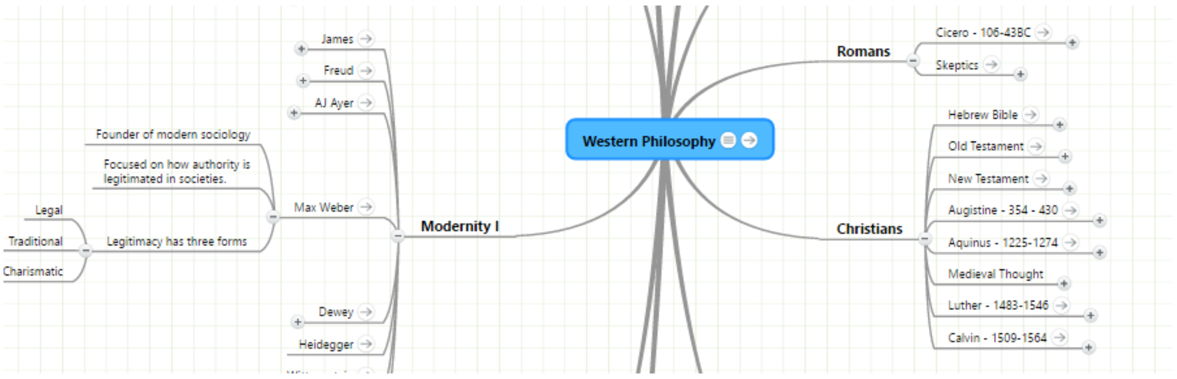


Figure 2.4: MindMeister interface

losing the overall view of the main subject.

2.1.2 Structured Argumentation Systems

In this section we will focus on structured argumentation systems. In Structured Argumentation systems, each argument is divided in different parts. For example, we can have some premises and infer some claim, all the parts together build an argument. Since the relations between every part of the argument are explicitly defined, the role of each part is also clearly defined.

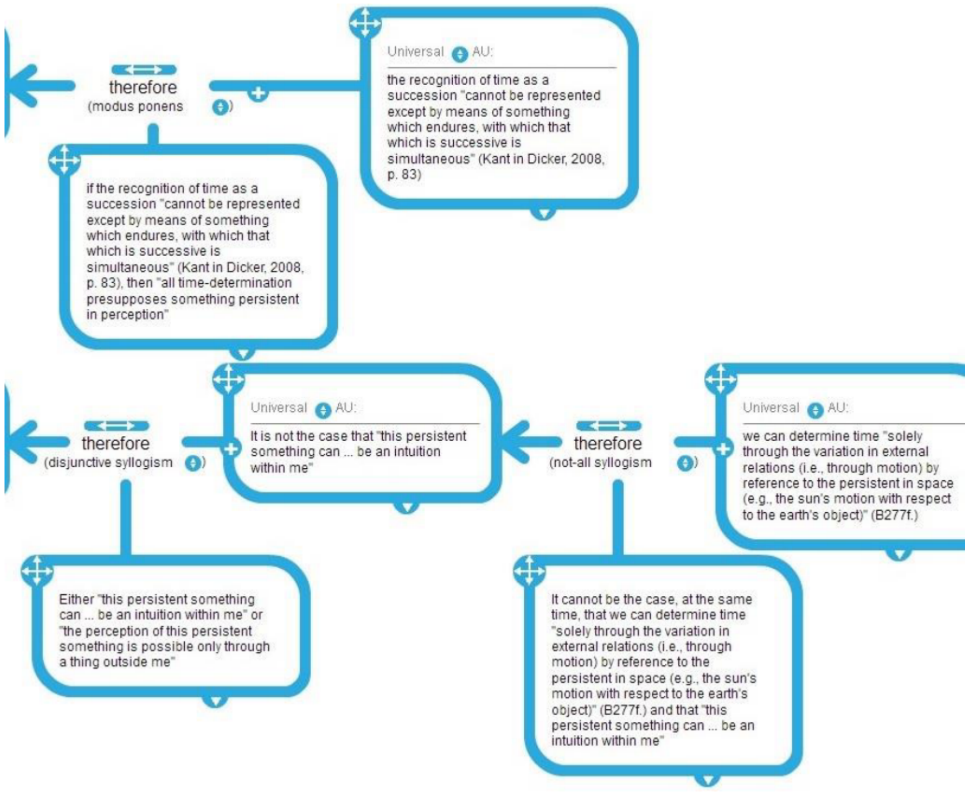
Let us look at the example of agora.gatech.edu[1], where the arguments are represented as nodes and the relations are represented as edges (figure 2.5). The role of each node is directly associated with the relation that it has with other nodes. If the relation between a node *a* and a node *b* is *"modus ponens"* we can conclude that *a* is a premise to conclude *b*.

To sum up, these kind of systems are used to structure and strengthen a thesis rather than to provide a open debate to the users.

2.1.3 Discussion

After analysing all these systems, we note that no existing system (that we are aware of) can fulfil all the requirements of a debate tool based on Social Abstract Argumentation. The applications that are based on abstract argumentation are the ones that are closest to what we intend, although there are a lot of points that are missing. These systems do not let the user introduce a new thesis inside a debate, furthermore as they are represented in threads, it is difficult to have the overall view of the debate.

Structured Argumentation based applications are further from what we aim with this thesis, they let the user establish a limited set of relations (e.g. cause/effect relation) between nodes narrowing the user's freedom to argue. On the other hand, the graph representation gives the user a view of the whole system and relations, that might be close to what we intend with our tool.



the recognition of time as a succession "cannot be represented except by means of something which endures, with which that which is successive is simultaneous" (Kant in Dicker, 2008, p. 83)

if the recognition of time as a succession "cannot be represented except by means of something which endures, with which that which is successive is simultaneous" (Kant in Dicker, 2008, p. 83), then "all time-determination presupposes something persistent in perception"

It is not the case that "this persistent something can ... be an intuition within me"

we can determine time "solely through the variation in external relations (i.e., through motion) by reference to the persistent in space (e.g., the sun's motion with respect to the earth's object)" (B277f.)

Either "this persistent something can ... be an intuition within me" or "the perception of this persistent something is possible only through a thing outside me"

It cannot be the case, at the same time, that we can determine time "solely through the variation in external relations (i.e., through motion) by reference to the persistent in space (e.g., the sun's motion with respect to the earth's object)" (B277f.) and that "this persistent something can ... be an intuition within me"

Figure 2.5: Structured Argumentation Systems: Agora-net

The mind maps are used to decompose ideas into smaller parts, they do not let a user participate, leading to an informative hierarchical graph. However, this kind of systems are not what we aim; however, they have some properties that can be used in our solution. Focus on smaller parts of a big discussion and organization of arguments into branches are some properties that might be useful to achieve our goals.

To conclude, we want to create a system that has different goals or properties in relation to the solutions discussed previously. The system should provide an easy way to interact and also have different mechanisms for users to provide feedback. The solution should also provide the users all the freedom to participate and give his/her opinion, via new arguments and voting.

2.2 Semantics for Abstract Argumentation

In this section, we will start by explaining what is an Abstract Argumentation Framework using the model proposed by Dung back in 1995 [19] and also *ranking-based argumentation semantics*. Afterwards we will explore some other relevant Abstract Argumentation semantics and their properties.

2.2.1 Dung's Abstract Argumentation Framework

All the defenitions throughout this section were first proposed and described in [19].

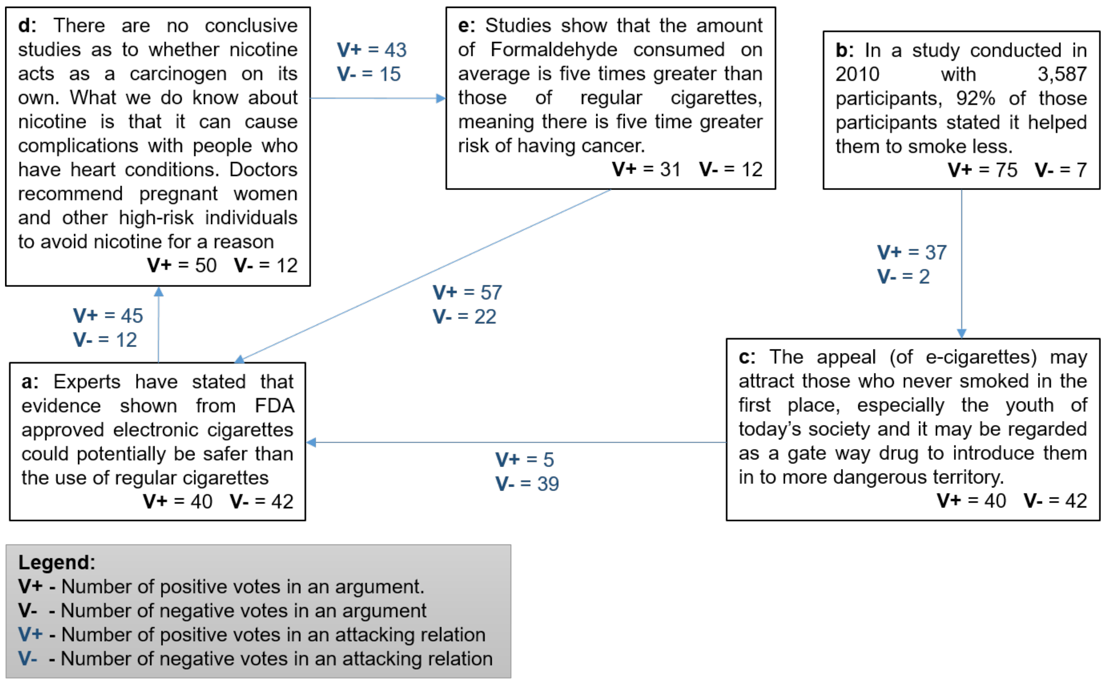

d: There are no conclusive studies as to whether nicotine acts as a carcinogen on its own. What we do know about nicotine is that it can cause complications with people who have heart conditions. Doctors recommend pregnant women and other high-risk individuals to avoid nicotine for a reason
V+ = 50 **V-** = 12
V+ = 43 **V-** = 15
e: Studies show that the amount of Formaldehyde consumed on average is five times greater than those of regular cigarettes, meaning there is five time greater risk of having cancer.
V+ = 31 **V-** = 12
b: In a study conducted in 2010 with 3,587 participants, 92% of those participants stated it helped them to smoke less.
V+ = 75 **V-** = 7
V+ = 37 **V-** = 2
V+ = 57 **V-** = 22
V+ = 45 **V-** = 12
a: Experts have stated that evidence shown from FDA approved electronic cigarettes could potentially be safer than the use of regular cigarettes
V+ = 40 **V-** = 42
V+ = 5 **V-** = 39
c: The appeal (of e-cigarettes) may attract those who never smoked in the first place, especially the youth of today's society and it may be regarded as a gate way drug to introduce them in to more dangerous territory.
V+ = 40 **V-** = 42
Legend:
V+ - Number of positive votes in an argument.
V- - Number of negative votes in an argument
V+ - Number of positive votes in an attacking relation
V- - Number of negative votes in an attacking relation

Figure 2.6: Graph Representation of a Social Abstract Argumentation Framework

Definition 2.1. An Argumentation Framework is a pair

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

where A is a set of arguments and R is a binary relation in A. The relation $R \subseteq A \times A$ defines an attack within A.

An Abstract Argumentation Framework can be represented by a directed graph where the nodes stand for arguments and the edges represent attacks. If some argument b attacks an argument c, where $b, c \in A$ and $(b, c) \in R$, we will have an edge directed from b to c (Figure 2.6). The AF represented in figure 2.6 (ignoring the votes) is defined as follows: $AF_{2.6} = \langle \{a, b, c, d, e\}, \{(a, d), (c, a), (d, e), (e, a), (b, c)\} \rangle$

Definition 2.2. A set S of arguments is said to be conflict-free if:

$$\forall a, b \in S : \neg \exists (a, b) \in R$$

Definition 2.3.

1. An argument $a \in A$ is acceptable to (defended by [5]) a set S if and only if:

$$\forall b \in A : (b, a) \in R \Rightarrow \exists x \in S : (x, b) \in R$$

2. A conflict-free set of arguments S is admissible if and only if:

$$\forall a \in S : a \textit{ is acceptable with respect to } S$$

Definition 2.4. A preferred extension of an argumentation framework AF is the maximal admissible set (with respect to set inclusion) of AF.

The notion of preferred extension is used to define the semantics of an argumentation framework.

Definition 2.5. Let S be a conflict-free set of arguments. S is called *stable extension* if and only if S attacks every argument that does not belong to S (i.e. $\forall a \in A \setminus S, \exists b \in S : (b,a) \in R$)

Definition 2.6. A set S of arguments is called a *complete extension* if and only if each argument, acceptable with respect to S, belongs to S.

Definition 2.7. Two Argumentation Frameworks $AF_1 = \langle A_1, R_1 \rangle$ and $AF_2 = \langle A_2, R_2 \rangle$ are isomorphic if and only if there is a bijective mapping $m : A_1 \rightarrow A_2$ such that:

$$(a,b) \in R_1 \Leftrightarrow (m(a), m(b)) \in R_2$$

2.2.2 Ranking-based Semantics for Abstract Argumentation

In this section a set of Ranking-based semantics for Abstract Argumentation will be compared. The ranking-based semantics classify each argument with a large range of levels of acceptability [10], producing a unique ranking between the arguments. As we imagine a platform meant for Social Abstract Argumentation, we envision a debate as a graph where nodes represent arguments and edges represent attacks. Both, edges and nodes, have negative and positive votes associated (see figure 2.6). The strength or acceptance of an argument will be defined by the relations and the votes only. The strength of an attack will be solely defined by its votes.

First, we will need to define some properties, used in [10], in order to compare the different semantics. Since this properties were not defined to a debate as we envision we will also mention what it is needed to be changed in order to use each property. Throughout this section we will define the direct attackers of a as $R^-(a)$ and the defenders (i.e. an attacker of an attacker of a) as $R^+(a)$.

Abstraction.[2] *The ranking on A should be defined on the basis of attacks between arguments.* This is a relevant property, however we must add that the ranking depends not on the content of the argument but on its attacks and votes.

Independence.[2]*The ranking between two arguments $a, b \in A$ should be independent of any other argument $c \in A$, if c is not attacked/attacks neither a nor b.*

Void Precedence.[2] *An argument that has not been attacked is ranked strictly higher than one which was attacked.* This is a relevant property, however it does not consider the number of votes of an argument. So it only stands if the strength of both arguments is the same and if at least one of the attacks (in the attacked argument) has its strength is not null.

Self-Contradiction. *An argument that attacks itself is ranked strictly lower than any other argument that does not attack itself.* This property can only be verified if the strength of all the attacks is the same as well as the difference between its negative and positive votes.

Cardinality Precedence.[2] *The greater the number of direct attackers of an argument, the weaker the level of acceptability it has.* We need to reformulate this property, in order to apply it to our concrete use case. If we have two arguments a, b where a is being attacked by n arguments and b by m, and $m > n$, a has a better level of acceptability than b, if and only if the strength of the attack relations and the attackers is equal for a and b, we also have to consider that both arguments, a and b have the same difference between negative and positive votes.

Quality Precedence.[2] *The greater the acceptability of an direct attacker of an argument, the weaker the level of acceptability of the argument.* This property does not consider the strength of the attack relation.

Counter-Transitivity.[2] *If the direct attackers of a are at least as numerous and acceptable as those of b, then b is at least as acceptable as a.* This property does not consider the strength of the attack relation, therefore it is only relevant if we consider that every attack in the example have the same strength.

Strict Counter-Transitivity.[2] *If the direct attackers of a are strictly more numerous or acceptable than those of b, then b is strictly more acceptable than a.* This property does not consider the strength of the attack relation, therefore it is only relevant if we consider that every attack in the example have the same strength.

Defend Precedence[2] *For two arguments $a, b \in A$ with the same number of direct attackers, if a is defended and b non-defended, then a will have a higher ranking than b.* This property does not consider the strength of the attack relation, therefore it is only relevant if we consider that every attack in the example have the same strength.

Distributed-Defend Precedence[2] *The best defense is when each defender attacks a distinct attacker.* This property is relevant if we consider that all the attack relations strengths are not null.

Total. *All pairs of arguments can be compared.* This property is relevant since, in order to make a ranking it is mandatory that every two arguments can be directly compared, even if they are not directly related.

Non-attacked equivalence. *All the non-attacked arguments must have the same rank.* This is a relevant property because, this property must be fulfilled, in order to keep a meaningful feedback to the user.

Other properties:

Properties related to adding/increasing an attack/defense branch. *An attack branch on a is a branch of arguments attacking its sibling where a is the leaf of the branch and the number of arguments is odd. A defense branch on a only differs to the attack branch on the total number of arguments, in this case we have an odd number of arguments in the branch.* These properties are not relevant to the solution since they do not take into account the weight of the attack (defined by votes).

2.2.2.1 Social Abstract Argumentation Framework

The Social Abstract Argumentation Framework[15, 21, 24] is an extension of Dung's AAF since it adds votes to the previous definition. The Social Abstract Argumentation Framework was first proposed considering only votes on the arguments. Later an extended version of the Social Abstract Argumentation Framework introduced the votes on attacks. The following definitions are related to the extension to the Social Abstract Argumentation Framework.

Definition 2.8. Let AF be a Social Abstract Argumentation Framework where $AF = \langle A, R, V_A, V_R \rangle$. $V : A \to \mathbb{N} \times \mathbb{N}$, is a total function mapping each argument (V_A) or attack(V_R) to its number of positive and negative votes.

Definition 2.9. A social abstract argumentation semantic framework is a 6-tuple $\langle L, \tau, \curlywedge_A, \curlywedge_R, \curlyvee, \neg \rangle$, where:

- L is a totally ordered set with top and bottom elements $\top, \bot$, containing all possible valuation for an argument.
- $\curlywedge_A, \curlywedge_R : L \times L \to L$, are two binary algebraic operations to restrict strengths to given values (where A stands for argument strengths, and R to attack strengths).
- $\curlyvee : L \times L \to L$, is a binary algebraic operation on argument valuations used to combine or aggregate valuations and strengths.
- $\neg : L \to L$, is a unary algebraic operation for computing a restricting value corresponding to a given valuation or strength.
- $\tau : \mathbb{N} \times \mathbb{N} \to L$, is a function that aggregates positive and negative votes into a social support value.

Definition 2.10. Let F be a social abstract argumentation framework and S a semantic framework, where $S = \langle L, \tau, \curlywedge, \curlyvee, \neg \rangle$. A total mapping $M : A \to L$ a social model of F under semantics S, or S-model of F, if:

$$M(x) = \tau(V(x)) \curlywedge \neg\curlyvee\{M(x_i) : x_i \in R^-(x)\}\ \forall x \in A$$[2]

This semantics consider the votes in the arguments to compute the strength of an argument as well as the votes on the attack relations. If one argument $a \in A$ has a high valuation, and it attacks $b \in A$, we can not guarantee that b will have a low valuation, since the votes on the relation $(a,b) \in R$ may decrease the strength of the attack to nearly 0, thus this attack will have almost no influence on the valuation of b.

After using these semantics, and the respective implemented algorithm[15], that had to be adapted to consider also the votes on attacks, to calculate the score of each argument in the graph represented in the figure 2.6, we got the following values for both versions SAA and SAA Extended — with votes on attack relations. The results are shown in table 2.1.

Table 2.1: Results for SAA and SAA Extended

	a	b	c	d	e
SAA Extended	0.342254	0.914634	0.064522	0.588548	0.406361
SAA	0.316293	0.914634	0.0416419	0.551377	0.323426

Table 2.1 shows that the votes on attack relation can have a huge impact on the score of an argument. For example, if we consider argument e it changes approximately 0.08 from SAA to SAA Extended.

2.2.2.2 Categoriser

The categoriser [6] is a function that assigns for each argument a value, given the value of its direct attackers. $Cat : A \rightarrow]0,1]$ is defined as follows:

$$Cat(a) = \begin{cases} 1, & \text{if } R_1^-(a) = \emptyset \\ \dfrac{1}{1 + \Sigma_{c \in R_1^-(a)} Cat(c)}, & \text{otherwise} \end{cases}$$

These semantics takes into account only the direct attackers of an argument to compute its rank. Since the Categoriser function takes into account the score of other arguments to calculate the score of an argument, it may lead to nonlinear equations that can be difficult to calculate and also may have more than one result per argument. Developing an algorithm to calculate this scores may not be a trivial task to do.

If we apply this function to the example in the previous subsection (Figure 2.6) we get: $Cat(a) \approx 0.477$, $Cat(b) = 1$, $Cat(c) = 0.5$, $Cat(d) \approx 0.677$, $Cat(e) \approx 0.596$.

[2] $\curlyvee\{x_1, x_2, ..., x_n\} \triangleq ((x_1 \curlyvee x_2) \curlyvee ... \curlyvee x_n)$. M(x) is referred as the social strength, or value, of x in M.

Table 2.2: Discussion-based semantics

step	a	b	c	d	e
1	2	0	1	1	1
2	-2	0	0	-2	-1

2.2.2.3 Discussion-based semantics

Discussion-based semantics [2] are based on a linear discussion. A linear discussion is a sequence of arguments such that each argument attacks the argument preceding it in the sequence. We define $Dis_i(a)$ as follows:

$$Dis_i(a) = \begin{cases} -|R_i^+(a)|, & \text{if i is odd} \\ |R_i^-(a)|, & \text{if i is even} \end{cases}$$

In these semantics the number of attacks to an argument is more important than the strength of the arguments that are attacking. When applying this semantics to the AF represented in figure 2.6, we get the results reported on table 2.2 from where we get the following ranking, that is ordered lexicographically: $b \succ_{DBS} d \succ_{DBS} e \succ_{DBS} c \succ_{DBS} a$.

2.2.2.4 Burden-based semantics

The Burden-based semantics [2] are based on Burden numbers. The Burden number of an argument a is defined based on the burden numbers of its direct attackers. We define $Bur_i(a)$ as follows:

$$Bur_i(a) = \begin{cases} 1, & \text{if} i = 0 \\ 1 + \Sigma_{b \in R_1^-(a)} \dfrac{1}{Bur_{i-1}(b)}, & \text{otherwise} \end{cases}$$

As on Discussion-based semantics the number of attacks to an argument is more important than the strength of the arguments that are attacking it. When applying this semantics to the AF represented in Figure 2.6, we get the results reported ontable 2.3 from where we get the following ranking: $b \succ_{BBS} d \succ_{BBS} e \succ_{BBS} c \succ_{BBS} a$.

Table 2.3: Burden-based semantics

step	a	b	c	d	e
1	3	1	2	2	2
2	2	1	2	1.33	1.5
3	2.17	1	2	1.5	1.75

2.2.2.5 Valuation with tuples

The valuation with tuples[13] takes into account all the ancestor branches of an argument stored in tupled values. Let $a, b \in A$, we consider the number of attack branches on a

and b, as well as the defense branches. Afterwards we compare those numbers to define whether $a \prec_{VT} b$ or $b \prec_{VT} a$.

Note that if the AF has cycles, some tuples can be infinite, to solve this, this algorithm transforms cycles into infinite acyclic graphs to calculate them. This is not a trivial algorithm to calculate, and as there was no implemented version of the algorithm, we will not show the results for this algorithm and we will only consider its main properties.

2.2.2.6 Matt & Toni

Matt and Toni [25] compute the strength of an argument based on a game. This game confronts two players, a proponent and an opponent of a given argument, where the strategies of a player are sets of arguments. Let S_P be the strategies for the proponent and S_O for the opponent, $S_O, S_P \subseteq A$.

Definition 2.11. The degree of acceptability of P with respect to O is given by

$$\phi(P,O) = \frac{1}{2}[1 + f(\text{N. of attacks from P to O}) - f(\text{N. of attacks from O to P})]$$

where $f(n) = \dfrac{1}{1+n}$

Definition 2.12. The rewards of P, denoted by $r_F(P,O)$, are defined by:

$$r_F(P,O) = \begin{cases} 0, & \text{iff } \exists a,b \in P, (a,b) \in R \\ 1, & \text{if N. of attacks from O to P} = 0 \\ \phi(P,O), & \text{otherwise} \end{cases}$$

Definition 2.13. For each argument $a \in A$ the proponent's expected payoff is given by:

$$E(a,p,q) = \Sigma_{j=1}^{n} \Sigma_{i=1}^{m} p_i q_j r_{i,j}$$

Where p and q are probability distributions with $p = (p_1, p_2, ..., p_m)$ and $q = (q_1, q_2, ..., q_m)$.

Lastly the value of the game is denoted by:

$$s(a) = max_p min_q E(a,p,q)$$

In Matt & Toni the strength of the attackers has more impact on the valuation of the arguments than the number of attackers. Note that this is not a trivial algorithm to calculate, and as there was no implemented version of the algorithm, we will not show the results for this algorithm and we will only consider its main properties.

2.2.3 Discussion

After analysing all these semantics we can reach the following conclusions. All the discussed semantics have the properties of Abstraction, Independence and Total. They all produce a partial ranking between the arguments and this ranking depends only on the votes and attacks of each argument. However *Tuples** does not guarantee the Total property, since there are some cases where two arguments can not be compared. All the other properties had to be reformulated in order to consider the strength of the attack relations as well as the votes on the arguments.

Some of the semantics discussed earlier are difficult to implement since this implementation would lead to much complex algorithms. The *Categoriser* semantics are not smooth and also they do not provide an implemented algorithm. Other algorithms such as *Discussion-Based Semantics* and *Burden-Based Semantics*, rely more on the number of the attacks rather than on the strength of the arguments and attacks. The *Matt & Toni* algorithm could also be a possibility however, it also has some limitations, for example the semantics are not smooth, this means that a small change in the graph could lead to great changes in the ranking.

To sum up, only one of the semantics presented, *Social Abstract Argumentation Framework*, allows the users to vote on arguments and relations. Furthermore, the semantics are smooth, since a small change in the state of the graph will never lead to a big change in the ranking between the arguments. Additionally, an efficient implementation of the algorithm is available. Table 2.4 presents the rankings for some of the semantics discussed throughout this section, in order to illustrate the impact of the votes in the ranking process.

After analyzing table 2.4 it is clear that the votes have a huge impact in the argument ranking, even for a small example with only five arguments, SAA Extended produces a different result from all the other semantics.

Table 2.4: Ranking Results Comparison

	1	2	3	4	5
SAA Extended	b	d	e	a	c
Categoriser	b	d	e	c	a
DBS	b	d	e	c	a
BBS	b	d	e	c	a

2.3 System Architecture

As we saw on the previous sections the system that we are trying to develop will have a potential large number of users, arguments and relations. In order to deal with this volume of data and computations, we will resort to a distributed architecture to materialize our system. Nevertheless, this system must be scalable, robust and have a good performance while providing ways of authentication and authorization. We must not forget that to calculate the score of a single node of the argumentation graph, we may need to recalculate the score of all the arguments or a large fraction of them in the graph and this needs to be done in a short period of time. To achieve this, we will examine three distinct technologies that may be able to address this problem.

2.3.1 MPICH-G: A grid-enabled implementation of the Message Passing Interface

The first technology that we are going to address in this section is based on the Message Passing Interface that was firstly proposed on the year 1992 [18]. We will examine an extension of this first idea, MPICH-G [22] that may fulfil the requirements needed for our solution.

MPICH-G has two main blocks: MPICH — one of the most widely use implementation of MPI — and Globus, a toolkit to build wide area applications. The MPICH block grants the system portability and performance , since the same syntax can be used over different architectures and, as it is a mature technology, its performance is already strongly optimized. The Globus block provides the system Authentication and Authorization as well as fault detection services, which are essential to ensure security and availability.

MPI allows a developer to design and implement a custom made parallel algorithm to address his application needs. With this programming model it is possible to have many processes working in parallel that communicate between them. However, this customization takes a lot of study to become optimal, therefore we will approach other possible solutions.

2.3.2 Incoop: MapReduce for Incremental Computations

The second programming model that we are going to approach is a model based on the MapReduce — a widely used system proposed by Jeffrey Dean and Sanjay Ghemawat from *Google, Inc.* [16]. Incoop is a MapReduce framework for incremental computations [8] that allows MapReduce programs, that were not initially designed for incremental processing, to execute in an incremetal manner. The performance of the computations may increase when using this framework since Incoop reuses intermediate results from the previous runs to recompute the desired values every time the data changes, in a more timely fashion.

2.3.2.1 MapReduce

MapReduce is one of the most used systems in distributed and parallel computing. This programming model expresses the computation as two functions:

Map. This function takes as input multiple key-value pairs and produces a set of intermediate of output key/value pairs. In the next step, all of this elements are grouped together using the intermediate key, and then they are passed to the reduce function.

Reduce. This function accepts an intemediate key and a set of values for that key. It then merges those values together by key in order to form a possibly smaller set of values. The intermediate values are supplied to the reduce function through an iterator.

MapReduce allows to have multiple instances running the *map* and *reduce* functions, using a master to coordinate all the workers, therefore it must tolerate machine failures gracefully.

Worker Failure. The master pings every worker periodically. If the worker does not answer in a certain amount of time, master marks the worker as failed. All the completed map tasks assigned to that worker have to be re-executed, however the same strategy does not apply to the reduce tasks since the output is not stored locally.

Master Failure. The master must write periodic checkpoints of its data structures. If the master dies a new copy can be started from the last checkpoint state.

2.3.2.2 Incoop

Incoop is a system that allows MapReduce programs to execute transparently in an incremental[3] manner [8]. Since in Inccop the itermediate results from previous runs are reused, the system can respond faster to changes in the input data. The technique used on Incoop relies on memoization, performing a stable partitioning of the input and reducing the granularity of tasks maximizing result reuse.

As to maximize the efficiency of the system, other mechanisms were developed:

Incremental HDFS. A file systems that identifies similarities in the input data of consecutive runs.

Contraction phase. A phase that would be used to control the granularity dividing the large task into smaller subtasks, maximizing the opportunity of reusing the result of a previously completed task.

Memoization-aware scheduler. A work stealing algorithm[4] to minimize the amount of data movement across machines.

To sum up this system can actually prevent the whole recomputation of a debate graph by using incremental computing. The input is divided into smaller parts and then it is distributed to different instances, that will run the map functions and reduce functions.

[3]The aim of incremental computation is to save time when recomputing outputs every time a piece of data changes [12]

[4]In work stealing algorithms processors needing work steal computational threads from other processors.[9]

Moreover, as we saw previously, this functions are optimized and can reuse previous results that are stored on a memoization server.

2.3.3 Discretized Streams

The last programming model that we will be addressing is based on the stream processing programming paradigm. The stream processing programming paradigm simplifies the use of parallel processing by the applications. This is simplified by applying a set of operations to a stream of data, where this set of operations is applied for each element of the stream. Therefore we can have each stream running in a different processor.
Discretized Streams[29] provides streaming computations as a series of deterministic batch computations on short time intervals.
The Discretized Streams works as follows:

1. Each time interval the input data is stored reliably across the cluster forming an input dataset for that interval.

2. When the time interval completes the dataset is processed using deterministic paralell operations (e.g map, reduce and groupBy).

3. New datasets are produced representing program outputs or intermediate states of the computation.

4. Results are stored in a resilient distributed datasets (RDDs)[5].

The system works with a group of resilient distributed datasets and provides multiple operators to manipulate them. The operators can be divided in stateful and stateless.

Stateless operators. Stateless operators, such as map, acts independently on each time interval.

Stateful operators. Stateful operators, such as aggregation over a sliding window, acts on multiple intervals and may produce intermediate RDDs as state.

In summary, this system provides the necessary tools to use incremental computation that can avoid the whole recomputation of a debate graph, hence granting a better performance to the algorithm. The dynamic processing of the data is also a plus, considering the kind of interactions that our tool will have, also every time an user interacts with the system that action is integrated in the stream, therefore providing a reactive output.

[5]Resilient Distributed Datasets (RDDs) are a distributed memory abstraction that lets programmers perform in-memory computations on large clusters in a fault-tolerant manner.[30]

2.3.4 Discussion

After studying these three options there are some measures that we must take into consideration. If we consider the performance measure, only Incoop and Discretized Streams have a out of the shelf implementation that can easily be integrated on our prototype. However, these two solutions have multiple and sgnificant differences, Incoop does not process the data dynamically, the data is divided across the different instances and is only computed afterwards. On the other hand, Discretized Streams process the data on demand, since it works with short time intervals and new data can be integrated into the next time interval, therefore the system becomes potentially more reactive.

All of the presented alternatives are scalable and provide features for Fault Tolerance as well as Authorization and Authentication mechanisms. To sum up, the best alternative seems to be Discretized Streams, since it provides all the properties that our system requires, while being readily available.

Chapter 3

Methodology

In this chapter we will start by describing the project requirements, system architecture and tools and technologies that we will use to develop our prototype. We then discuss alternatives for experimentally evaluating our prototype, and finally, we present a schedule for the remainder of the work to be conducted and a summary of the document.

3.1 Project Requirements

In this section we will define the project requirements, divided in functional and non-functional requirements.

3.1.1 Functional Requirements

1. The user must be able to add a new argument.

2. The user must be able to add a new attack relation between arguments.

3. The user must be able to vote positively or negatively on a relation or on an argument.

4. The debate must be presented at least in the form of a graph.

5. The user must distinguish the different weights of each relation and argument.

Other functional requirements such as user authentication may also be considered throughout the development of this thesis.

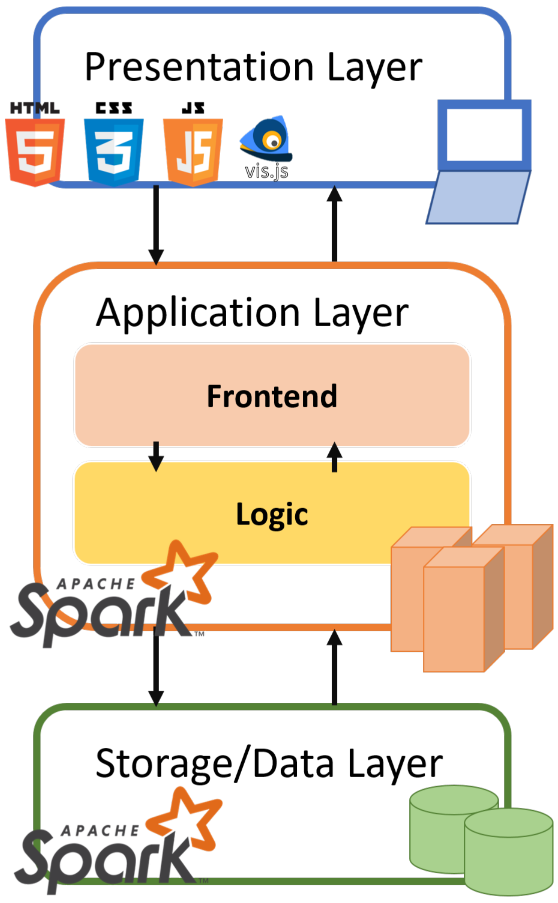


Figure 3.1: Three-tier architecture

3.1.2 Non-functional Requirements

1. The system must provide a good performance. The results of changing data should be visible to the user with low latency.

2. The system should be tolerant to faults. If a machine of the infrastructure is down the system must grant availability. If all the infrastructure is down, the system must be able to recover to its last state after the recovery of the infrastructure.

3. It should be easy to integrate the tool into an existing website.

3.2 System Architecture

Our framework architecture will be based on the Three-tier model firstly proposed by Eckerson in 1995 [20] (figure 3.1).

Presentation Layer. The Presentation Layer is the topmost level of the application. In our solution the presentation layer will be the system interface along with the requests to the application layer. This layer will also deal with all the information that comes from the application layer. Furthermore we will communicate with the application layer using RESTful communications.

Application Layer. The application layer will be hosted in various machines (e.g. cluster). This part of the tool will deal with all the logic behind it (e.g. semantics algorithm) and it will also retrieve the data from the storage layer. It is divided into two sublayers, Frontend and Logic. The Frontend will be hosted on a web server to provide a web plugin that can be embedded in a website. The logic layer will deal with all the computations and requests to the storage layer.

Storage/Data Layer. This layer stores all the data that it is needed to maintain the application, such as users, arguments, debates and relations data. This data will be stored in the same machines that are used in the application layer.

3.3 Tools & Technologies

In order to build a prototype we have to use the appropriate technologies. In this section we will describe which technologies we are going to use for each part of the system. Figure 3.1 illustrates our system architecture along with some of the technologies that are going to be used.

3.3.1 Presentation Layer

To create the interface of the plugin we intend to use the library Vis.js[11] that allows visualization, manipulation, and interactions with data represented through a graph where both edges and nodes are customizable (see figure 3.2). The rest of the interface will be developed using HTML5, CSS, and Javascript and also some libraries that may be necessary (e.g. libraries to communicate using rest/AJAX).

3.3.2 Application Layer

This layer will be developed on Apache Spark[3].The frontend and logic sublayer of the application layer must provide RESTful web services in order to communicate with the upper layer. This can be achieved using technologies such as the Jersey Rest Framework.

Algorithm. The algorithm will be implemented in *Java* in order to run on Apache Spark. The algorithm is currently implemented in C++, thus it must be reimplemented in Java. Moreover, other adaptations will be required, to enable the parallelization of the algorithm.

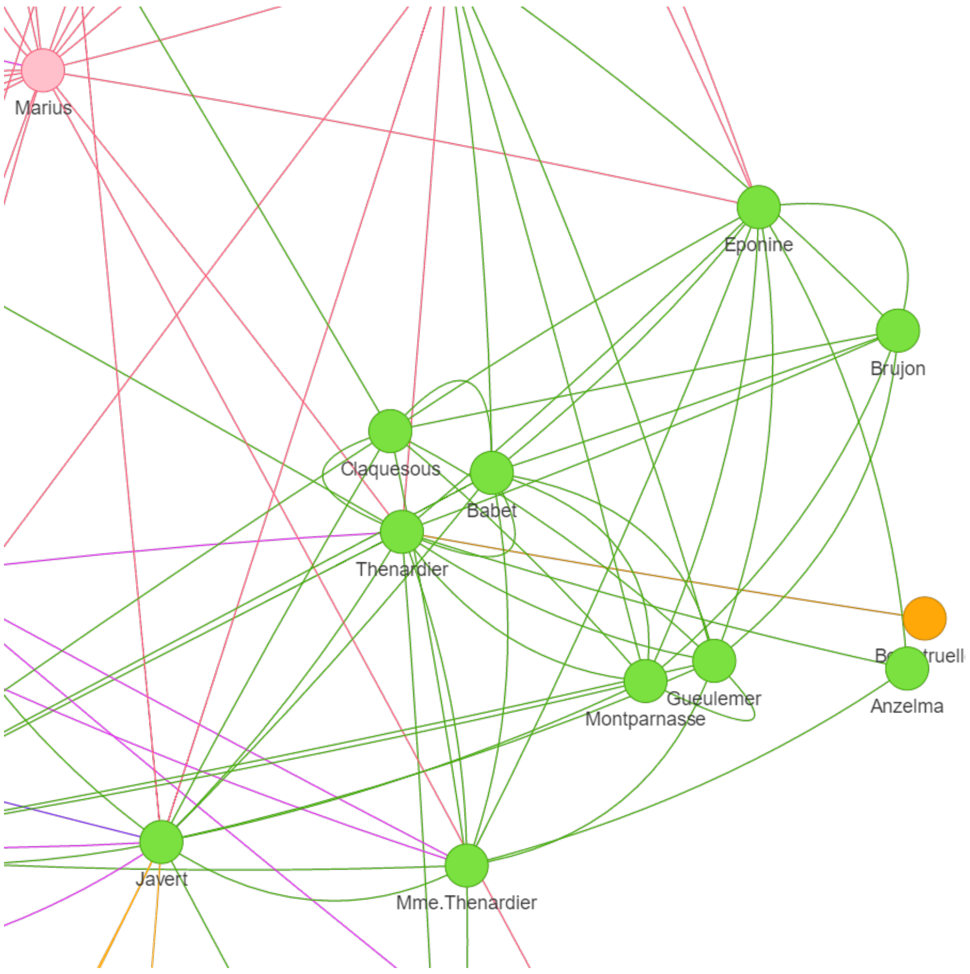


Figure 3.2: Vis.js interface example

3.3.3 Storage Layer

Apache Spark[3] storage is granted by RDDs that work under Hadoop File System. The Hadoop Distributed File System provides scalability to the system. Apache Spark also provides a collection of operators to manipulate and manage the RDDs.

3.4 Evaluation Plan

Our project should fulfil all the requirements described in section 3.1. In order to evaluate it, we will describe the metrics that we are going to use in the remainder of this section.

3.4.1 Ranking-based Semantics for Abstract Argumentation

To test the correctness of the algorithm we will use a synthetic workload with some study cases for which we already know the results, and check if the results match.

3.4.2 Interface for an Online Debate Tool

To test the interface we will make user tests with predefined scenarios, where each user will have to complete a determined number of tasks and after answer a small questionnaire about their experience. Some of these tests will be done under observation in order to better diagnose the users' difficulties while interacting with the system and adjust the interface accordingly. The system does not have to be fully operational in order to do these tests, only a prototype with a working interface is necessary.

3.4.3 System Architecture

To evaluate the system architecture we need to take some performance measures such as throughput and latency. In order to do this we will use a heavy, and possibly or typically generated, workload. We also need to test the system's tolerance to faults. To achieve this we will use the same workload but we will shutdown some instances and check if the system is still available and working normally, as well as measure the impact of these faults in the overall system performance.

3.4.4 System Evaluation

After the individual evaluation of all the previous points, we will have to evaluate the system as a whole. To accomplish this, we will do some additional user tests different from the ones that we did previously. One of the tests that we intend to do will be with a group of people that we will encourage to start a debate, from scratch, using our system. Afterwards we will see how the debate evolves and we will also try to identify additional problems.

3.5 Project Schedule

We defined the following tasks to be completed throughout the MSc dissertation:

1. Environment Setup
2. Algorithm Adaptation
3. Features
    - 3.1. Add a new argument
    - 3.2. Add attacks between arguments.
    - 3.3. Vote on an argument
    - 3.4. Vote on a relation
    - 3.5. Graph Visualization
    - 3.6. Nodes and Edges customization

4. Evaluation and Testing
    4.1. Algorithm Testing
    4.2. User Testing
    4.3. Performance Testing
    4.4. Fault Tolerance testing
5. System Evaluation
6. Writing

The project schedule is organized with respect to the features of the project. There are some tasks that do not depend on each other, the feature of Graph Visualization will be developed along with all the other features except from the nodes and edges customization.

Figure 3.3 illustrates the amount of time assigned to each task.

Although the evaluation phase is separated from the implementation of the features, some of the evaluation tasks can be completed during the implementation. The rest of the tasks must be completed in the order defined in figure 3.3 since they depend on each other.

3.6 Summary

In this document we presented the problem of the actual debate tools and what we propose as a solution. The proposed solution was an online debate tool that can be integrated in a website as a plugin. To achieve this solution we have studied the existing applications, semantics and architectures to approach our problem. In the last chapter we have described the methodology that we are going to use, defining which technologies and tools we will be using and the metrics to evaluate our system.

	February				March				April				May				June				July				August				September			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Environment Setup					🟦	🟦	🟦	🟦																								
Algorithm Adaptation								🟧	🟧	🟧	🟧																					
Features												🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩												
1. Add a new argument												🟩	🟩																			
2. Add attacks between arguments														🟩	🟩	🟩																
3. Vote on an argument																🟩	🟩															
4. Vote on a relation.																🟩	🟩															
5. Graph Visualization												🟩	🟩	🟩	🟩	🟩	🟩															
6. Nodes and Edges customization																		🟩	🟩	🟩												
Evaluation and Testing																		🟨	🟨	🟨	🟨	🟨	🟨	🟨								
Algorithm testing																		🟨	🟨													
User testing																				🟨	🟨	🟨										
Performance testing																				🟨	🟨											
Fault tolerance testing																						🟨	🟨									
System Evaluation																						🟪	🟪	🟪	🟪	🟪	🟪					
Writing												🟥	🟥					🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥

Figure 3.3: Project Schedule

Bibliography

[1] *Agora-net. Collaborative and web-based argument visualization software that can be used for free all over the globe.* URL: http://agora.gatech.edu.

[2] L. Amgoud and J. Ben-Naim. “Ranking-based semantics for argumentation frameworks”. In: *International Conference on Scalable Uncertainty Management*. Springer. 2013, pp. 134–147.

[3] Apache. *Apache Spark^{TM}. Lightning-fast cluster computing.* URL: http://spark.apache.org/.

[4] ARG-tech. Centre for Argument Technology. URL: http://www.arg-tech.org/.

[5] P. Baroni and M. Giacomin. “Semantics of Abstract Argument Systems”. In: *Argumentation in Artificial Intelligence*. Ed. by G. Simari and I. Rahwan. Boston, MA: Springer US, 2009, pp. 25–44. ISBN: 978-0-387-98197-0. DOI: 10.1007/978-0-387-98197-0_2. URL: http://dx.doi.org/10.1007/978-0-387-98197-0_2.

[6] P. Besnard and A. Hunter. “A logic-based theory of deductive arguments”. In: *Artificial Intelligence* 128.1 (2001), pp. 203 –235. ISSN: 0004-3702. DOI: http://dx.doi.org/10.1016/S0004-3702(01)00071-6. URL: http://www.sciencedirect.com/science/article/pii/S0004370201000716.

[7] P. Besnard, A. Garcia, A. Hunter, S. Modgil, H. Prakken, G. Simari, and F. Toni. “Introduction to structured argumentation”. In: *Argument & Computation* 5.1 (2014), pp. 1–4.

[8] P. Bhatotia, A. Wieder, R. Rodrigues, U. A. Acar, and R. Pasquin. “Incoop: MapReduce for Incremental Computations”. In: *Proceedings of the 2Nd ACM Symposium on Cloud Computing*. SOCC ’11. Cascais, Portugal: ACM, 2011, 7:1–7:14. ISBN: 978-1-4503-0976-9. DOI: 10.1145/2038916.2038923. URL: http://doi.acm.org/10.1145/2038916.2038923.

[9] R. D. Blumofe and C. E. Leiserson. “Scheduling multithreaded computations by work stealing”. In: *Journal of the ACM (JACM)* 46.5 (1999), pp. 720–748.

[10] E. Bonzon, J. Delobelle, S. Konieczny, and N. Maudet. “A Comparative Study of Ranking-based Semantics for Abstract Argumentation”. In: *CoRR* abs/1602.01059 (2016). URL: http://arxiv.org/abs/1602.01059.

[11] A. B.V. *vis.js. A dynamic, browser based visualization library.* 2016. URL: http://visjs.org/.

[12] M. Carlsson. "Monads for Incremental Computing". In: *SIGPLAN Not.* 37.9 (Sept. 2002), pp. 26–35. ISSN: 0362-1340. DOI: 10.1145/583852.581482. URL: http://doi.acm.org/10.1145/583852.581482.

[13] C. Cayrol and M.-C. Lagasquie-Schiex. "On the acceptability of arguments in bipolar argumentation frameworks". In: *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*. Springer. 2005, pp. 378–389.

[14] Consider.it. *Consider.it. Creates CIVIL, ORGANIZED, AND EFFICIENT ONLINE DIALOGUE by visually summarizing what your community thinks and why.* URL: https://consider.it/.

[15] M. Correia, J. Cruz, and J. a. Leite. "On the Efficient Implementation of Social Abstract Argumentation". In: *Proceedings of the Twenty-first European Conference on Artificial Intelligence*. ECAI'14. Prague, Czech Republic: IOS Press, 2014, pp. 225–230. ISBN: 978-1-61499-418-3. DOI: 10.3233/978-1-61499-419-0-225. URL: https://doi.org/10.3233/978-1-61499-419-0-225.

[16] J. Dean and S. Ghemawat. "MapReduce: Simplified Data Processing on Large Clusters". In: *Commun. ACM* 51.1 (Jan. 2008), pp. 107–113. ISSN: 0001-0782. DOI: 10.1145/1327452.1327492. URL: http://doi.acm.org/10.1145/1327452.1327492.

[17] *DebateGraph. To change the world you need to look at it in a different way.* URL: http://debategraph.org.

[18] J. J. Dongarra, R. Hempel, A. J. Hey, and D. W. Walker. *A proposal for a user-level, message passing interface in a distributed memory environment*. Tech. rep. Oak Ridge National Lab., TN (United States), 1993.

[19] P. M. Dung. "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games". In: *Artificial Intelligence* 77.2 (1995), pp. 321 –357. ISSN: 0004-3702. DOI: http://dx.doi.org/10.1016/0004-3702(94)00041-X. URL: http://www.sciencedirect.com/science/article/pii/000437029400041X.

[20] W. W. Eckerson. "Three tier client/server architectures: achieving scalability, performance, and efficiency in client/server applications". In: *Open Information Systems* 3.20 (1995), pp. 46–50.

[21] S. Eğilmez, J. Martins, and J. Leite. "Extending social abstract argumentation with votes on attacks". In: *International Workshop on Theorie and Applications of Formal Argumentation*. Springer. 2013, pp. 16–31.

[22] I. Foster and N. T. Karonis. "A Grid-enabled MPI: Message Passing in Heterogeneous Distributed Computing Systems". In: *Proceedings of the 1998 ACM/IEEE Conference on Supercomputing*. SC '98. San Jose, CA: IEEE Computer Society, 1998, pp. 1–11. ISBN: 0-89791-984-X. URL: http://dl.acm.org/citation.cfm?id=509058.509103.

[23] A. J. Freeley and D. L. Steinberg. *Argumentation and debate*. Cengage Learning, 2013.

[24] J. Leite and J. Martins. "Social abstract argumentation". In: *IJCAI*. Vol. 11. Citeseer. 2011, pp. 2287–2292.

[25] P.-A. Matt and F. Toni. "A game-theoretic measure of argument strength for abstract argumentation". In: *European Workshop on Logics in Artificial Intelligence*. Springer. 2008, pp. 285–297.

[26] mindmeister. *Mindmeister. Collaborative mind mapping.* URL: https://www.mindmeister.com/.

[27] J. Nielsen. *Usability Engineering*. Interactive Technologies Series. Morgan Kaufmann, 1994. ISBN: 9780125184069. URL: https://books.google.pt/books?id=95As2OF67f0C.

[28] TidyLifeInc. *CreateDebate - a social tool that democratizes the decision-making process through online debate.* URL: http://www.createdebate.com/.

[29] M. Zaharia, T. Das, H. Li, S. Shenker, and I. Stoica. "Discretized Streams: An Efficient and Fault-tolerant Model for Stream Processing on Large Clusters". In: *Proceedings of the 4th USENIX Conference on Hot Topics in Cloud Ccomputing*. HotCloud'12. Boston, MA: USENIX Association, 2012, pp. 10–10. URL: http://dl.acm.org/citation.cfm?id=2342763.2342773.

[30] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica. "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing". In: *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*. USENIX Association. 2012, pp. 2–2.