7CCSMPRJ - Individual Project
Argupedia

Final Project Report

Abstract

This thesis describes the development of Argupedia - an innovative and ever-growing repository for contemporary debates. Unlike rival public argumentation platforms, Argupedia does not succumb to polarisation and echo-chambers - instead structuring knowledge graphically in order to expertly depict how arguments, opinions and points of view relate to each other and evolve over time.

Within Argupedia, users arguments and counter-arguments have to conform to schemes and critical questions which act as templates - ensuring that the knowledge dialogued between parties is valid and conform to principled theories of argumentation. Lastly, arguments submitted to Arguepdia can be expertly evaluated using labelling algorithms - impartially evaluating the status of the arguments submitted.

Argupedia is inspired by the works of leading AI argumentation specialists and academics such as Dung [22] and Walton [76]. Furthermore, the successfully constructed Argupedia prototype was initially user-experience (UX) tested in a pilot study versus a control and leading rival platform, namely Debate.org. Key results from the pilot study in chapter 12 find that:

- Four out of four (100%) participants self-reported that they learnt more about the philosophies of argumentation, logic and debate having used Argupedia versus the control, Debate.org.
- Argupedia outperformed Debate.org across most UX Attrakdiff questionnaire categories - scoring significantly higher in hedonic qualities and stimulation, meaning that Argupedia is both an engaging and innovative platform [30].
- Despite some minor front-end usability quirks limiting Argupedia's overall pragmatic quality score, users reported that they found Argupedia an "amazing concept" and a more "constructive debate platform" versus Debate.org.

The main contribution of this thesis is the Argupedia application itself coupled with UX feedback and results from testing the platform. However, additional contributions to the key research areas of AI argumentation include:

- Insights from Dung's abstract argumentation framework in chapter 2, argument schemes and critical questions in chapter 3.
- A thorough analysis of GUI design and HCI principles in chapter 4 coupled with a critical evaluation of rival public argumentation platforms in chapter 5.
- A thorough analysis and critique of the specification, tools and methodology for building Argupedia in chapters 7 to 9.
- An analysis of experiment design and the test protocol employed for UX testing Argupedia in a pilot study in chapter 10.
- Reflections on Argupedia as well as showcase argumentation graphs in chapter 11.

Originality Avowal

I verify that I am the sole author of this report, except where explicitly stated to the contrary. I grant the right to King's College London to make paper and electronic copies of the submitted work for purposes of marking, plagiarism detection and archival, and to upload a copy of the work to Turnitin or another trusted plagiarism detection service. I confirm this report does not exceed 15,000 words.

Acknowledgements

I would like to first and foremost thank the essential help and support of my supervisor Dr Sanjay Modgil. His advice, thoughts and meetings helped immensely in successfully executing the project and writing this thesis - I always learnt a huge amount from his insights. Also, I would like to thank my participants for sacrificing their time and offering exemplary feedback for this research. Additionally, I would like to thank the support of my parents for their unwavering help and allowing me to work at home relatively easily throughout the pandemic. Lastly, I would like to thank Carys Williams for just about everything else and many constructive discussions throughout the project.

Contents

Chapter 1

Introduction

1.1 Background

Argumentation is old and widespread - central for millennia as a tool for the accrual of human knowledge and a core concept of philosophy [23]. Most notably, the ancient Greek sophists specialised in utilising the tools of formal public debate, rhetoric and argumentation to both entertain and teach large crowds [51].

With the landmark innovations of the computer, world wide web and social media - it was inevitable that argumentation and debate communication flows would instead primarily occur online harnessing exceedingly different tools to the ancient Greeks. With the inevitable rapid progression in computer technology, so have the possibilities for tools of argumentation and debate increased significantly - as well as the size of the audience and extensive influence of public dialogues and discourses [66].

Equally, in recent years argumentation has become increasingly central as a core study within Artificial Intelligence (AI) [7]. The interdisciplinary field spanning logic, philosophy, cognitive science and communication studies offers many exciting pathways [77]. Two particularly effective pathways include firstly, improving the existing debate that occurs between human agents online and secondly, using public human debates to inculturate and regulate the activities of artificially intelligent agents [48, 77].

In contrast, at present much of the argumentation and debate occurring online is unconventional, irregularly structured, quite limited and in the wake of *Fake news* scandals sometimes deliberately fallacious [66]. Indeed, debates that occur on the web typically offer little to no formal analysis of the status of debates, evaluation of the existing arguments presented, with the proper use of schemas and arguments forged on the basis of critical questions.

Consequently, there is therefore a pressing need to overcome these concerning limitations and offer new modes of expression for debaters online. Insights from AI argumentation effectively used can thus serve to not supplant existing modes of debate and argumentation online, but rather augment and enrich them.

1.2 Argupedia and the problem domain

As initially noted above in section 1.1, at present there are several online debating platforms. However, the current online platforms suffer from three significant problems: firstly, there is no way in which members of the public are guided to structure arguments in a rational way, secondly members of the public are not guided guided in determining which types of argument are appropriate for challenging another persons argument and thirdly, the current platforms do not provide a formal way of evaluating the status of arguments - in effect, which arguments are winning, losing and more preferred.

This thesis describes the development of a new public and user-friendly argumentation and debate tool called Argupedia - pictured below with figure 1.1, which serves to overcome

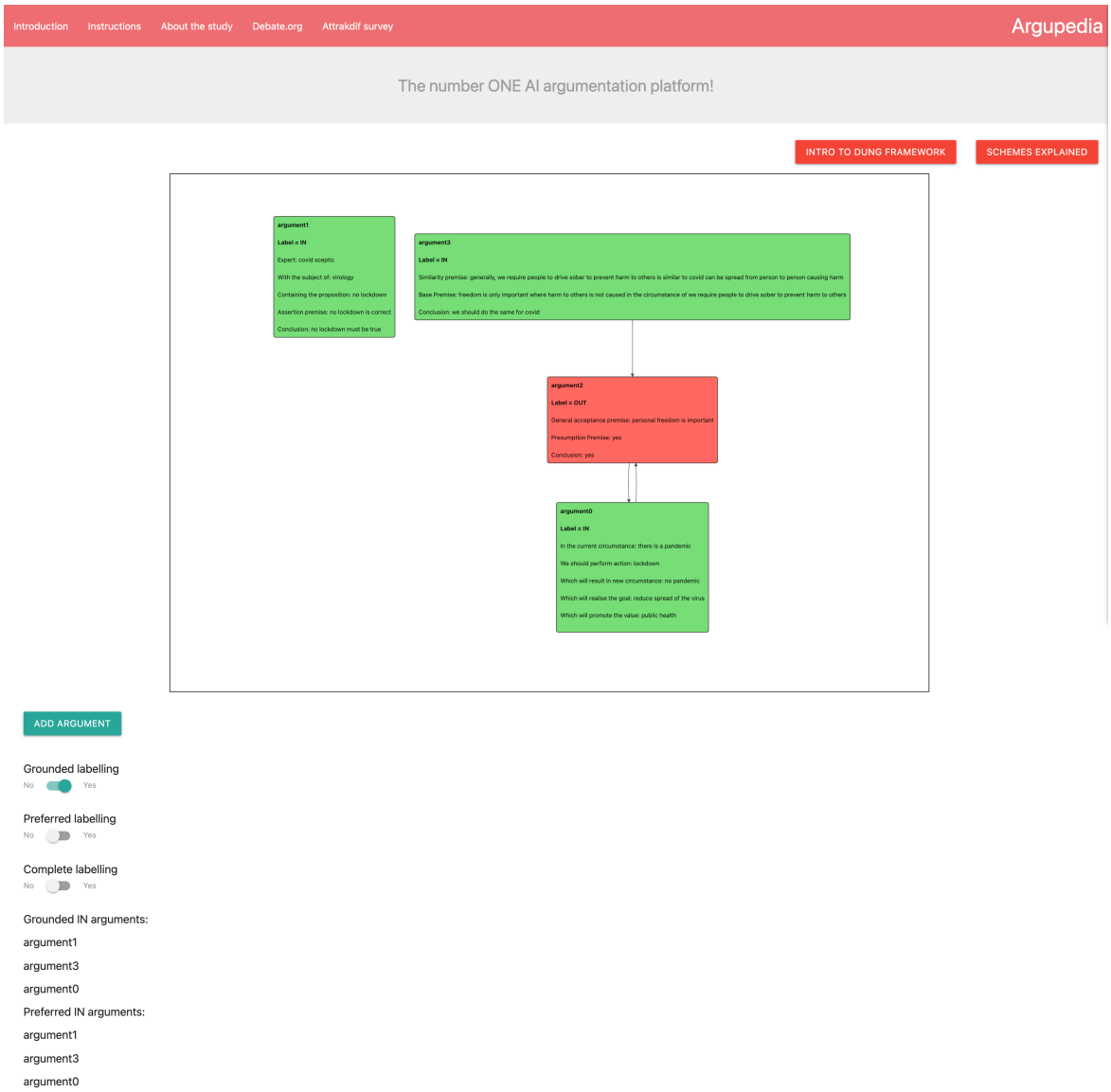


Figure 1.1: Screenshot of final Argupedia prototype with grounded labelling switched on.

these concerns. Argupedia is the first debate web application within the literature to offer a new means of argument and debate capture in a very different form from current domains online. Indeed, Argupedia offers a fully interactive graphical user interface (GUI) in which debaters can present different arguments according to predefined argumentation schemes and critical questions [79]. Therefore, the schemes serve as general templates from which users can construct arguments.

Heavily influenced by the framework of AI argumentation researcher Phan Minh Dung, after lodging initial arguments - the debaters can then proceed to counter-argue from which the application will automatically generate appropriate critical questions [22]. Before the user can once again use schemes as templates to construct counter-arguments. Having lodged valid counter-arguments, the application will sketch directed argumentation graphs in the form of nodes and edges [22].

Before, lastly the application harnesses grounded and preferred labelling algorithms - to label the graph and reveal the nodes and argumentation winners within the debate [49]. Over time, the application will provide a repository of debate an opinion which conforms to principled theories of argumentation - coherently capturing how public debates evolve.

1.3 Motivation

Despite the strong foundations of argumentation, argument and debate encountered within everyday contexts and echo-chambers rarely applies the traditional rigorous norms of argumentation and reasoning. Indeed, knowledge and information dialogued between human parties is too often incomplete, irrationally structured and fallacious [48]. Indeed, this problem is all too often compounded when public debate online centres on difficult topics such as ethics, meta-ethics and morality - these branches of philosophy are bridled with conjecture.

Additionally, as argued by D'Ancona and others, the ancient Greeks preference for debate and argumentation has markedly shifted from education syllabuses in favour of modern approaches centred upon memorisation and active recall of information - resulting in members of the public lacking cognitive awareness and the ability to argue to achieve logically valid conclusions [17].

In response to these concerns, the core motivation of Argupedia is to support a growing repository of logically coherent public argumentation graphs created by its user-ship concerning any issues worth debating. Utilising argumentation schemes and critical questions - users can provide attacking and counter-attacking arguments. From this, the application will be able to generate Dung argumentative graphs in which the interactive GUI can certify the set of winning or admissible arguments using different labelling algorithms [49]. Argupedia therefore greatly augments and enriches public debate occurring online.

As a contribution in a growing area of research, this thesis aims to support further work in this area by documenting the development process meticulously and by demonstrating online videos of use which may support and scaffold further research.

1.4 Project map

The thesis divides into four parts:

I. **Literature review:** Here I discuss the context for Argupedia and taxonomise related academic contributions.

   In chapter 2, I situate Argupedia within Dung's seminal theory of argumentation and compartmentalise the literature providing the most essential insights for Argupedia.

   Whilst, in chapter 3, I focus on the academic background of argument schemes and critical questions for structuring users arguments on Argupedia.

   Lastly in chapter 4, I discuss key GUI design and HCI principles to make Argupedia usable and robust as well as providing analysis of literature concerning UX testing.

II. **Specification, design and prototyping:** In this section the chapters focus on Argupedia's groundwork, context, experimental design and specification.

In chapter 5 I discuss other publicly available argumentation platforms before settling on Argupedia's unique selling point (USP).

In contrast chapter 6 focuses on the hypotheses for Argupedia's UX testing and key prototyping of the web application.

Finally, in chapter 7 I focus on outlining the requirements and specification of the Argupedia project.

III. **Implementation and development:** Within this section, I discuss the process of developing the Argupedia application via outlining the tools used and methodologies for both developping the app, running tests and successfully experimenting.

In chapter 8, I outline the tools used in this project.

Whilst in chapter 9, I discuss the process of designing and developing the Argupedia application and outline its pertinent features.

In contrast, in chapter 10, I discuss the experimental design - Attrakdif surveys and A/B UX testing. Before describing the white box versus black box strategies of testing employed in development.

IV. **Outcomes and evaluation:** In this section I describe the formal summative analysis and evaluation of the final Argupedia application. In results, I showcase argumentation graphs and consider useful future directions for work.

For chapter 11, I evaluate the final prototype and present some insightful graph examples generated with the Argupedia prototype.

Whilst in chapter 12, I present and analyse the results of the Argupedia prototype during UX testing trials as well as feedback from participants.

In chapter 13 I summarize findings, discuss future directions and provide concluding remarks.

1.5 Project outputs

The main contribution of this thesis is the Argupedia application itself - equipped with all necessary features to be usable for the general public. Ancillary contributions of the thesis include:

- A critique of the development process, tools and human computer interaction (HCI) graphical user interface (GUI) research used to build Argupedia.
- An evaluation of the Argupedia application on the basis of a pilot study - including UX feedback and suggestions for improving the platform, deducing the effectiveness of the platform and enabling pathways for future research.
- Collecting argumentation graphs which can support a growing repository of novel future AI argumentation research.

Chapter 2

Dung's abstract argumentation

This chapter situates Argupedia within Phan Minh Dung's seminal theory of argumentation and abstract argumentation framework. In this chapter I firstly, provide an overview of Dung's approach which inspired many new insights referred to as abstract argumentation [22].

Rather than providing an exhaustive literature review of all research regarding Dung's abstract argumentation framework - I focus on the literature most pertinent for developing and understanding the purposes of building the Argupedia web application.

2.1 Overview of abstract argumentation

The first part of Dung's paper outlines his theory of argument attack whilst in the second part he evaluates his theory via demonstration of two applications [22]. In brief, Dung's abstract argumentation framework is simply a pair (A, R) consisting of a set A whose elements are called arguments and of a binary relation R on A called an attack relation [61]. The set of arguments A is typically finite - with respect to the Argupedia application the set size ultimately depends upon the amount of arguments instantiated by users.

This argumentation framework can then be usefully visually represented as a directed graph where nodes are arguments and edges are drawn from attacking to attacked arguments - for reference see figure 2.1. The figure has the circular nodes represented as arguments whilst attacks between the arguments are registered as black arrows. As exemplified here, Dung's framework provides a very useful argument abstraction because he deliberately ignores the internal structure of arguments [69].

Indeed, Dung's abstraction is so profound that he does not provide a strict definition of an argument within the 1995 paper - traditionally an argument may be termed as a "line of reasoning leading from a premise to a conclusion" or perhaps "utterance in dispute" [22, 61] - ultimately Dung's abstract argument framework finds that an argument to be anything that may attack or be attacked by another argument [22].

Benefits of Dung's landmark abstraction is that it is highly accessible and integral for giving the freedom to build an interactive application like Argupedia. Indeed, abstracting away from the structure and meaning of arguments prevents formalism's from emerging which could be complex to implement in an interactive application. Especially if the argumentation application is designed to be primarily utilised by non-specialist members of the public.

Another concept discussed by Dung is the notion of an "admissible set of arguments" [22] - this can be understood as set of arguments validated by the dialogue and attack relations. Dung theorised that the set of arguments is admissible if it is *conflict free* and *acceptable* [22, 69]. In other words, a set of arguments is admissible if it contains no conflicts and can defend itself against all attacks [69]. Critically, these sets can be incorporated into the Argupedia application to successfully identify argumentation winners.

Although the principle of Dung's set of admissible argumentation seems straightforward enough - slightly more intricate puzzles can arise. As exemplified by figures 2.2 and 2.3 [12, 69]. figure 2.2 contains of two arguments: $\{A, B\}$ whereby each of the arguments are both admissibly

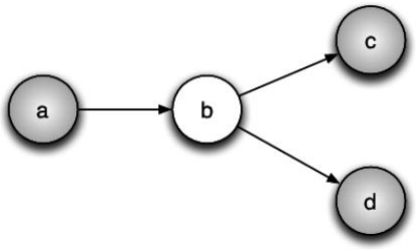


Figure 2.1: Here, the argument framework is formalized as $AF_1 = (A_1, R_1)$ where $A_1 = \{a, b, c, d\}$ and $R_1 = \{(a, b), (b, c), (b, d)\}$ [72]

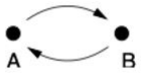


Figure 2.2: The bidirectional 'Nixon diamond' of attacks above contains two arguments $A_1 = \{a, b\}$ and $R_1 = \{(a, b), (b, a)\}$ [12]

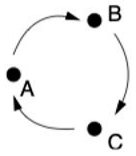


Figure 2.3: The cycle of attacks above contains three arguments $A_1 = \{a, b, c\}$ and $R_1 = \{(a, b), (b, c), (a, c)\}$ [12]

provable and refutable. Similarly in Figure 2.3 contains three arguments: $\{A, B, C\}$ yet none of the arguments are admissibly provable or admissibly refutable [69]. In effect, the three arguments effectively cancel out each others admissibility.

Due to the limitations of Dung's original labelling and sets of admissibility as exemplified by figures 2.2 and 2.3 useful framework extensions have been proposed by Dung and other academics to the abstract argument framework which will be discussed in the following section 2.2.

2.2 Framework extensions

In response to the limitations of Dung's framework discussed at the end of the previous section 2.1, an initial framework extension proposed by Dung was the notion of a *preferred* extension of an argument framework. The preferred extension makes the admissible set of arguments as large as possible - without adding elements that makes the set not admissible [22, 70].

To exemplify the preferred extension, in figure 2.2 there are two preferred extensions of the 'Nixon Diamond' - {a} and {b}. Whilst, figure 2.3 has just the one preferred extension the empty set. An extension to the preferred extension provided by Dung is the notion of a *stable* extension - these are defined as conflict-free sets that attack each argument not in the set.

The concepts of preferred and stable extension of an argumentation framework can be regarded as different ways to interpret a framework and thus they are often referred to as *semantics*. Argupedia will utilise these extensions or semantics to evaluate sets of arguments and give a status of the debate. Providing users with an integral understanding of the arguments that are winning, losing and most preferred. Another two important extensions proposed by Dung includes the *grounded* and *complete* semantics [22].

Grounded semantics is a concept that has its root originally in Pollock's OSCAR and the semantics of logic programming. A grounded extension is conflict free and more strict than the preferred extension because only the smallest element amongst the complete extensions [22]. From this, figures 2.2 and 2.3 would both have empty sets under the grounded extension or semantics.

Following Dung's paper [22] several kinds of semantics have been additionally proposed - for a thorough taxonomy please refer to Baroni et al. [5]. The concepts of the grounded, preferred and complete are utilised by Argupedia to provide labellings of the admissible set of arguments within a debate and further clarified in the following section 2.3.

2.3 Labelling arguments and algorithms

Argupedia will use labelling algorithms to express the acceptance of arguments and provide insights for users into the complete, grounded or preferred semantics set of arguments represented within a debate [22]. Within Dung's framework a labelling is a mapping that associates every argument with a label IN (the argument is accepted), OUT (the argument is rejected) or UNDEC (the argument is undecided - not accepted or refused).

Utilising the semantics of acceptance noted in the previous section 2.2, Argupedia will utilised the complete, grounded and preferred extensions to successfully label all arguments within a debate. Argument labellings provide an intrinsically essential purpose of the Argupedia platform because they enable the gamification of debate and can provide transparency to users over the currently most admissible arguments [49].

A detailed explanation with many examples of the complete, preferred and grounded labellings is provided by Modgil and Caminada accessible here: [49]. In brief and synthesising the paper, Modgil and Caminada define some critical ground rules which can provide a very clear understanding of argument labellings:

- An argument is labelled IN iff all its defeaters are labelled OUT.
- An argument is labelled OUT iff all it has at least one defeater that is labelled as IN.

- Grounded semantics minimise the set of IN labelled arguments and maximises the set of UNDEC labelled arguments under any complete labelling.
- Preferred semantics maximises the set of arguments that are labelled as IN under any complete labelling.
- For every preferred extension is also a complete extension.

Rounding off this chapter with a useful example, the following figures 2.4 and 2.5 contain just 5 arguments $A = \{a_1, a_2, a_3, a_4, a_5\}$ and $R = \{(a_1, a_2), (a_2, a_1), (a_3, a_1), (a_1, a_4), (a_4, a_1), (a_2, a_5), (a_5, a_2)\}$. In this case, there exists 1 grounded labelling and 2 preferred labellings - with arguments a_2 and a_5 taking all three possible labellings of UNDEC, IN and OUT whilst the other arguments remain: a_1 as OUT with a_3 and a_4 as IN.

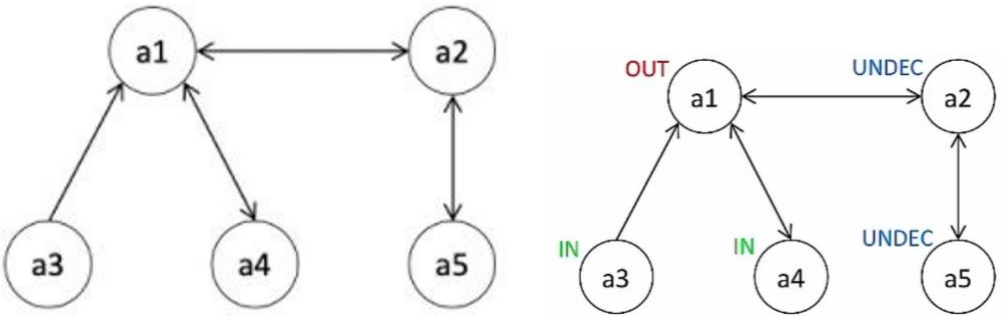


Figure 2.4: Labelling the left directed argument graph under the grounded semantics the right. Here a_3 would be labelled as IN as all arguments attacking the node are OUT. Consequently a_1 would be labelled as OUT as it has at least one argument attacking the node labelled as IN, resulting in a_4 being labelled as IN. Meanwhile A2 and A5 can equally be OUT or IN consequently they must be respectively labelled as UNDEC. The set of acceptable arguments under the grounded labelling is thus a_3 and a_4

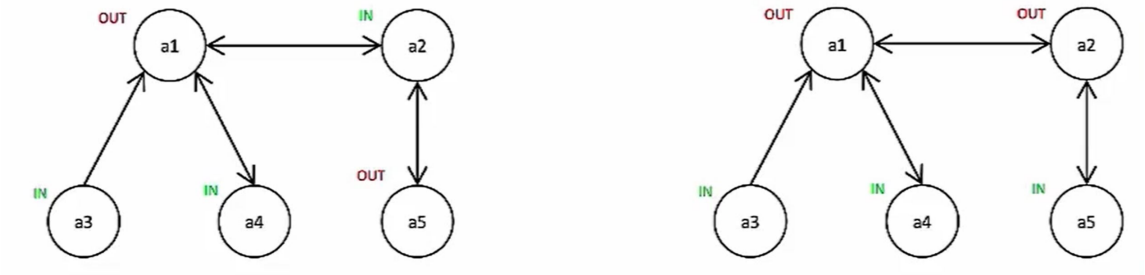


Figure 2.5: Labelling the graph under the preferred semantics results in two graphs. The labelling is the same as the grounded semantics for arguments a_1, a_2 and a_3. However, arguments a_2 and a_5 are labelled vice versa once IN and once OUT respectively this is because the preferred semantics is trying to maximise the number of argument nodes labelled as IN. Here, the set of acceptable arguments under the preferred credulous is larger as: a_2, a_3, a_4, a_5

Chapter 3

Argument schemes and critical questions

Rational argumentation and debate will only emerge effectively from Argupedia users if they are taught to structure arguments in a rational and effective way with meaningful reasons. In this chapter, I firstly discuss the foundation and establishment of argument schemes with a particular focus on influential AI argumentation research from Douglas Walton [74, 76].

Having provided initial insights into argument schemes - I will then briefly turn to the academic literature concerning critical questions which are critical to evaluate each type of argument [74]. Before finally focusing on the academic literature concerning the argument schemes and critical questions which will be directly employed within the Argupedia prototype.

3.1 The establishment of argument schemes

Branching originally from argumentation theory [57, 59], argument schemes can be though of as analogues of the rules of inference of classical logic for example *modus ponens* [70]:

P.
If P, then Q.
Therefore Q.

Whereas logical rules of inference such as the above modus ponens are abstract and strict, argument schemes are considerably more concrete and context dependent - in effect by Walton's definition they serve to capture stereotypical patterns of human reasoning [70, 74]. Indeed, the basis of these schemes arise from studying many examples of arguments - then deducing the patterns and structure common to these arguments [79]. In this sense, argument schemes will educate Argupedia users to structure their arguments premises to logically and validly achieve their preferred conclusions [71].

Derived from logic, argument schemes call upon sensible patterns that frequently occur in ordinary argumentation and formalise their structure [71, 74]. They are tremendously useful for a variety of fields from legal logic [81] to even medical reasoning [25]. Nonetheless, they can be misused resulting in the argument becoming defeasible i.e., the conclusion does not follow from the premises or contingent i.e., the scheme is used in the wrong context [71].

Within the academic literature argument schemes have a very long history - Aristotle listed common forms of argument which was the first systematic attempt to give an account of argumentation schemes [2, 62]. In the 20th century, Perelman and Olbrechts-Tyteca employed argument schemes properly as tools for analyzing and evaluating arguments [57, 74].

Following this, Hastings provided the first systematic analysis of common argumentation schemes including notably providing critical questions for each scheme [31, 74]. Recent further classifications of argumentation schemes include an extensive account developed by Kienpoint-

ner - who included deductive and inductive schemes. Lastly the most important and extensive analysis of ninety-six common presumptive argument schemes was conducted by Walton, Reed and Macagno [39, 79].

With respect to Argupedia, argument schemes are integral to the application because Argupedia will enforce users to utilise argument schemes as a template - guiding them to structure their arguments in a rational and logical way. The use of argument schemes will dually: educate Argupedia users and prevent them from submitting potentially fallacious and illogical arguments on the platform. To further exemplify these points about argument schemes for Argupedia, it is helpful to first read a definition of critical questions in section 3.2 before noting the schemes used by Argupedia in section 3.3.

3.2 Understanding critical questions

Critical questions as applied to everyday conversational arguments, are key devices that can be used to pinpoint potential weaknesses in a given argument [78]. Typically they are very useful in analysing a proponents argument and providing the potential for logical and effective rebuttals. Very usefully, critical questions can be generated dynamically as for each scheme argumentation researchers have already established the most effective critical questions [78].

In this respect, critical questions will server a very important purpose for Argupedia as they will enable users to confront, assess and counter-argue against arguments generated via schemes by other users on the platform [79]. Therefore critical questions enable users to independently perform assessments of what arguments are actually worth and whether to accept them - they are excellent for framing and evaluating arguments [78].

Having briefly discussed critical questions, now I will turn to discussing the schemes and critical questions to be specifically deployed in Argupedia - with a focus on the schemes academic context, advantages and use cases.

3.3 An overview of Argupedia's argument schemes and critical questions

Argupedia will deploy a significant variety of schemes and critical questions - for the initial prototype the aim is to utilise up to eight schemes in total sourced from: [3, 4, 16]. Focusing on the user testing, about eight schemes is certainly the optimum number for two reasons because: (a) Argupedia users will not be restrictive with too few schemes to liberally construct arguments of their choosing and (b) at the same time without being overwhelmed with too much unfamiliarity and choice. Now I will one by one discuss the academic foundations of each argument scheme to be employed by Argupedia and defend the use of the scheme against potential objections.

3.3.1 Critical action scheme

The Action-based Alternating Transition System (AATS) or critical action scheme was formalised by Atkinson and Bench-Capon to apply practical reasoning concerning moral problems [3, 4]. Origination from Hare's moral philosophy AATS seeks to enable sufficient reasoning about moral norms and principles on the basis of contextual situations [29].

The AATS argument scheme AS1 extends Walton's sufficient condition scheme for practical reasoning [76] and is structured in figure 3.1. The substantial sixteen critical questions certainly interrogate all *prima facie* justifications of proposals for action and is noted in figure 3.2. Furthermore, Atkinson and Bench-Capon even effectively go a step further by providing a good example of how to deploy the scheme using a state transition diagram of agents as automata in figure 3.3.

Potential criticisms of AATS may centre on the fact that it fails to enable interlocutors to effectively reason with actual moral norms, questions and principles directly. However, particularly for users of Argupedia, that line of moral reasoning can be difficult and ineffective.

AS1 In the current circumstances R
We should perform action A
Which will result in new circumstances S
Which will realise goal G
Which will promote value V.

Figure 3.1: AS1 from the AATS argument scheme [3].

CQ1: Are the believed circumstances true?
CQ2: Assuming the circumstances, does the action have the stated consequences?
CQ3: Assuming the circumstances and that the action has the stated consequences, will the action bring about the desired goal?
CQ4: Does the goal realise the value stated?
CQ5: Are there alternative ways of realising the same consequences?
CQ6: Are there alternative ways of realising the same goal?
CQ7: Are there alternative ways of promoting the same value?
CQ8: Does doing the action have a side effect which demotes the value?
CQ9: Does doing the action have a side effect which demotes some other value?
CQ10: Does doing the action promote some other value?
CQ11: Does doing the action preclude some other action which would promote some other value?
CQ12: Are the circumstances as described possible?
CQ13: Is the action possible?
CQ14: Are the consequences as described possible?
CQ15: Can the desired goal be realised?
CQ16: Is the value indeed a legitimate value?

Figure 3.2: Critical questions from the AATS argument scheme [3].

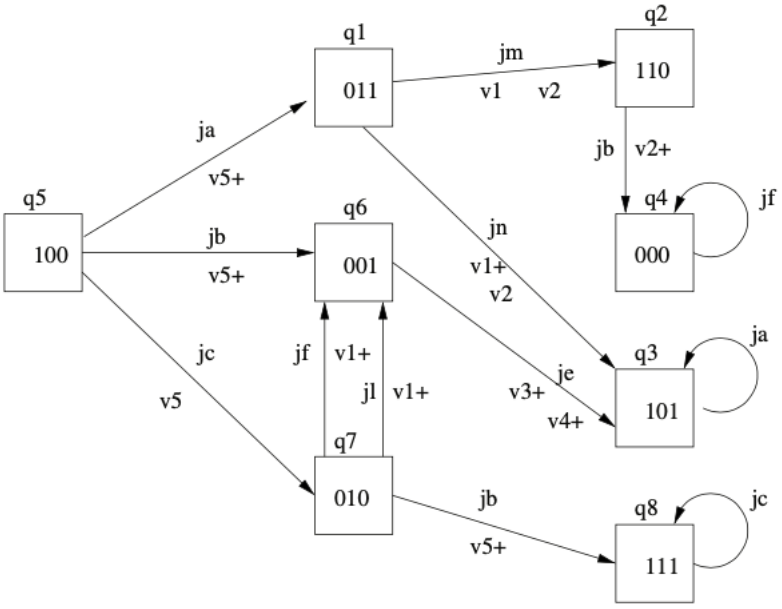


Figure 3.3: State transition diagram of agent under AATS argument scheme [3].

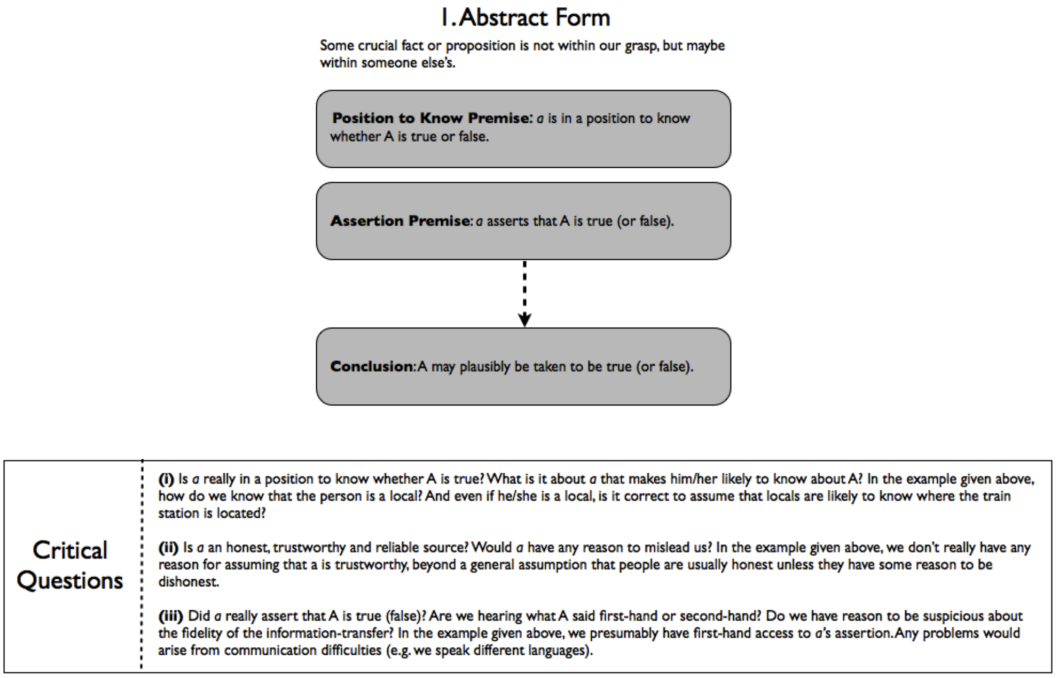


Figure 3.4: Argument from the position to know scheme and critical questions [16].

Therefore AATS favours moral reasoning on the basis of features of distinct scenarios. Indeed, this approach in which morality is crystallized in a not too specific or detailed form on the basis of situations can be a much more effective approach.

3.3.2 Argument from the position to know

Developed by Walton [76, 79], argument from position centres on epistemological principles. It is structured in a very similar way to the earlier defeasible *modus ponens* rule, which is formalised by many systems of non-monotonic logic [60]. The scheme is structured with critical questions as follows in figure 3.4.

A potential criticism of the scheme is that its conclusion is anything but definitive and instead only "plausible". However, this criticism fails to understand the purpose of argument schemes. Argument schemes are abstract inference schemes each with their own ways of being critically tested - they are not meant to be logically definitive.

3.3.3 Appeal to expert opinion

An early formulation of the appeal to expert opinion came from Walton in: [75]. To successfully apply the argument scheme requires identification of an expert within a subject domain voicing a specific proposition. The scheme uses the following structure and has six critical questions to evaluate the argument scheme as demonstrated in figure 3.5.

A weakness of the scheme is that it is potentially quite unnatural for interlocutors to use for conversational arguments. Yet, this weakness is purely as a consequence of the fact that the scheme was originally developed for the legal domain such as forensic scientists testifying evidence [78].

3.3.4 Appeal to popular opinion

Once again formulated by Walton, appeal to popular opinion has just two critical questions which are denoted in figure 3.6. A critique of the scheme is that *ad populum* are typically

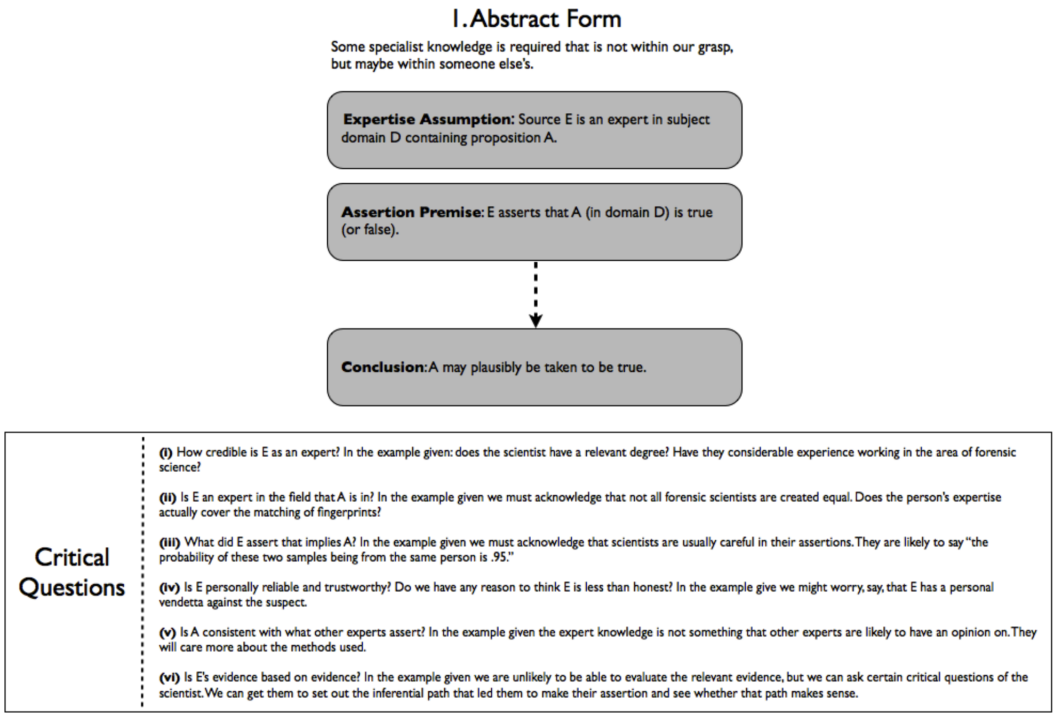


Figure 3.5: Appeal to expert opinion scheme [16].

considered as outright fallacious [24]. However, Walton successfully defends appeal to popular opinion effectively by offering a systematic historical review of the argument, its effectiveness and importance in debates [76].

3.3.5 Argument from analogy

A well-versed and effective form of reasoning - argument from analogy is structured with three critical questions in the following figure 3.7. In particular, Juthe offers a totally comprehensive analysis of the argument noting its frequent use in debate and discourse [36].

3.3.6 Argument from correlation to cause

The structure of argument from correlation to cause scheme and critical questions are denoted in figure 3.8. This scheme has been heavily employed within the scientific domain and to prevent logical fallacy is typically used to infer causation under *very* strict conditions, due to the fact that there could remain an at-present unobserved relation between the two variables [20]. Fortunately, the third critical question successfully highlights this potential vulnerability in the use of the argument.

3.3.7 Argument from consequences

Penultimately, argument from positive or negative consequences uses this abstract form and four critical questions for the scheme are noted in figure 3.9. Hoeken et al., offer a thorough and deep analysis of this form of argumentation and its psychological effectiveness. There analysis intriguingly found that this argument is most effective when utilised as a form of persuasion referring to a positive outcome rather than a dissuasive negative outcome [33].

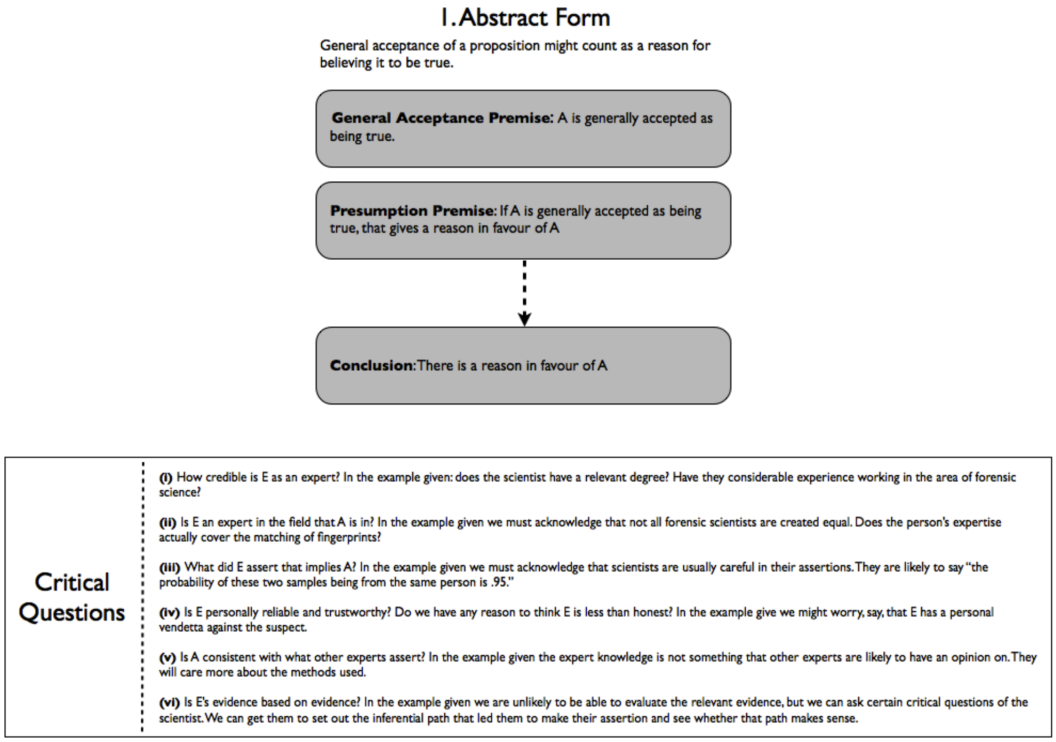


Figure 3.6: The appeal to popular opinion scheme [16].

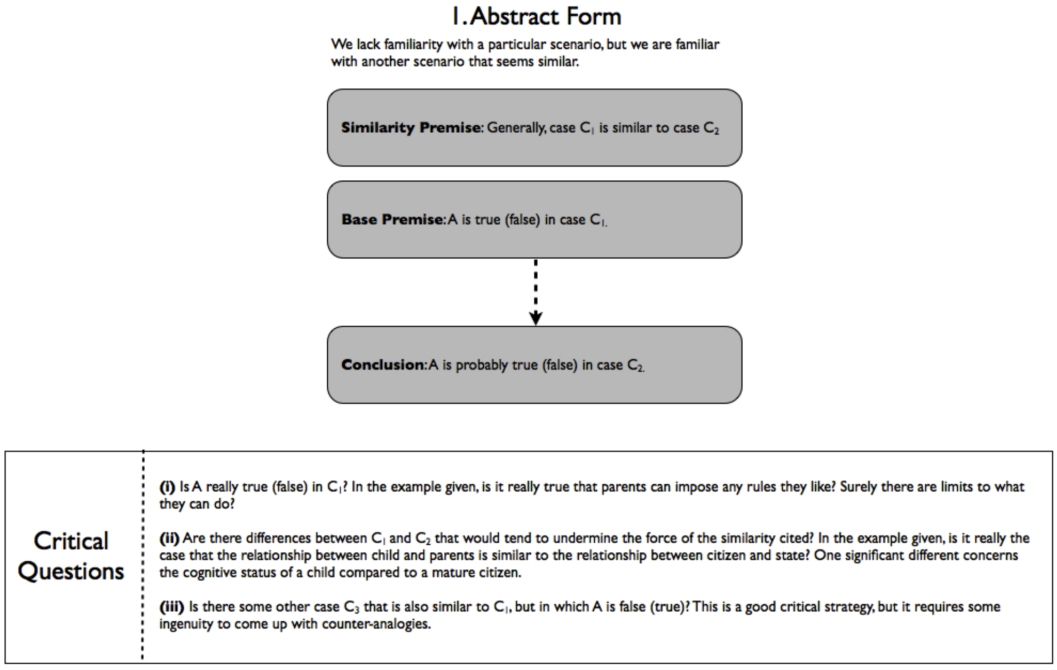


Figure 3.7: Argument from analogy and critical questions [16].

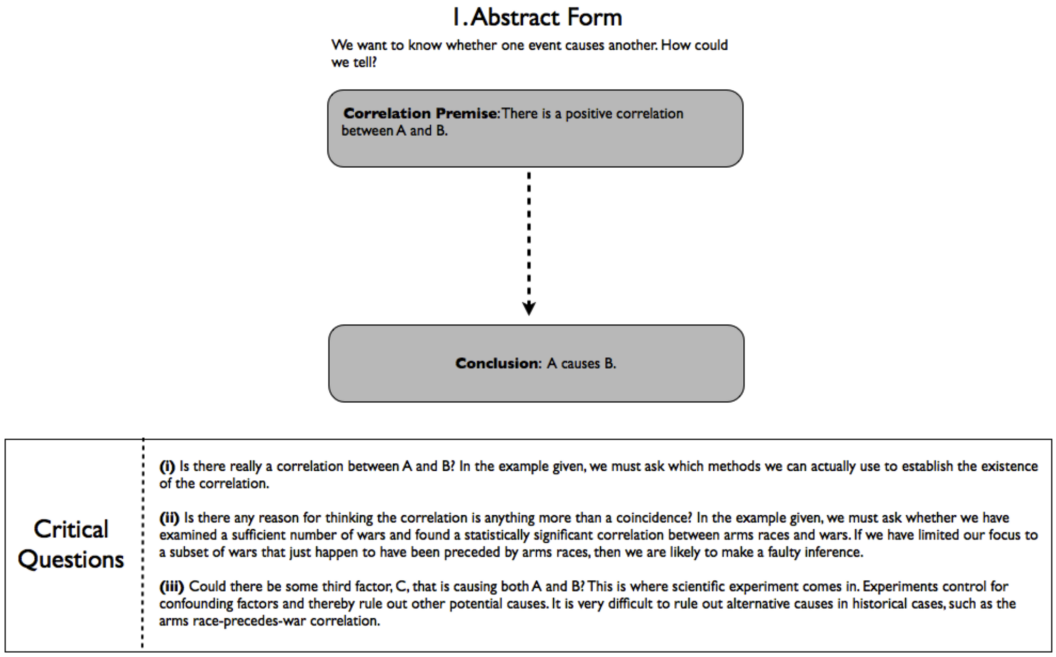


Figure 3.8: Argument from correlation to cause [16].

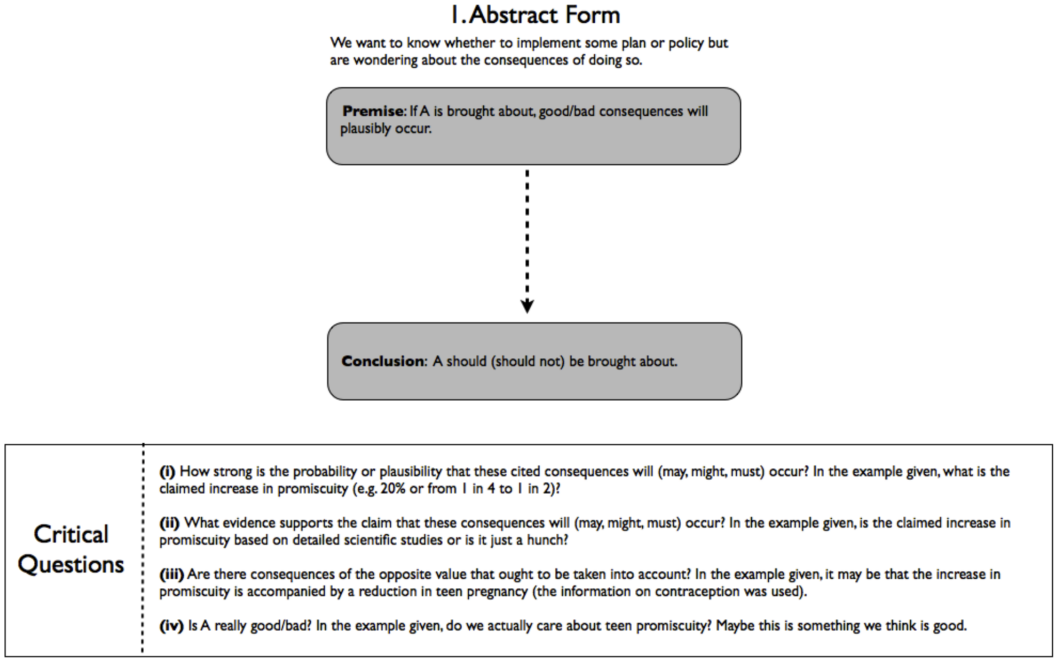


Figure 3.9: Argument from consequences [16].

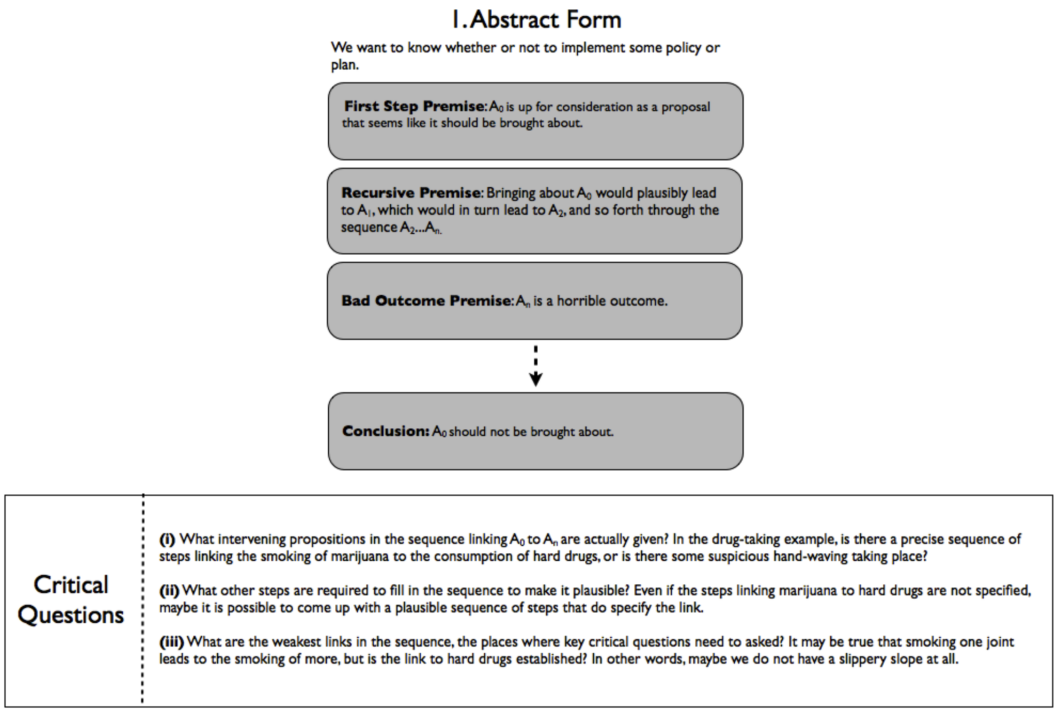


Figure 3.10: The slippery slope argument and critical questions [16].

3.3.8 The slippery slope argument

Finishing this chapter on Argupedia's schemes and critical questions, the slippery slope argument uses the abstract scheme developed by Walton [73] and three critical questions denoted in figure 3.10.

As noted by Walton and Burgess, this form of argument can very easily be employed in a logically deficient manner on highly contentious issues [11, 73]. However, both acknowledge that if employed correctly the slippery slope arguments are well-used and often integral for shifting the burden of proof and thus advancing critical discussion [73]. Argupedia will ensure this thanks in large part to critical questions clearly evaluating the effectiveness of the output argument and presenting pathways for counter-argument.

Chapter 4

GUI design & HCI principles

Argupedia is a complex and deeply academic research project - however the end users will almost certainly *not* always be experts in the fields of philosophy and AI argumentation. Consequently, this chapter focuses on HCI academic literature that outlines how Argupedia can be most usable with users requiring a very minimal cognitive load whilst contributing and using the open source platform.

In this chapter, I will initially focus on user interface design and Nieslsen's ten heuristics. Before turning to the literature insights on benefits of good interface design. Thirdly, I will discus academic insights from the engineering design process. Lastly, I will define and evaluate literature concerning user experience (UX) testing.

4.1 User interfaces and heuristics

User interface design is a fast growing subset of the field of human computer interaction. Specifically, the user interface is the part of the computer and its software that users can see, hear, touch, understand and direct - inputting and communicating their needs [26]. In the case of Argupedia, the graphical interface will be a web page - with users interacting through a screen utilising their desktop, laptop, tablet or phone.

Oppermann, Gallitz and many other academics within the field have all boldly asserted that the user interface is the most significant part of a software product [26, 55]. With respect to the project, this claim is certainly true as Argupedia's graphical user interface is integral for users to learn and use the system effectively (achieving their goals), efficiently (using limited resources) and acceptably (the user is satisfied) [9, 55].

Nielsen's heuristics - in effect, principles for good design will be used to maximise the effectiveness of the Argupedia interface [53]. The primary advantage of Nielsen's heuristics as pictured in figure 4.1 is that they offer a logical, thorough and empirical guideline for designing any interface - ensuring the interface maximises effectiveness, efficiency and acceptability [53, 54]. Additionally, any interface designed using Nielsen's heuristics can be greatly improved using professional evaluation to effectively locate any usability problems [54]. Now, I will discuss some of the benefits of Argupedia having a well designed interface.

4.1.1 Benefits of good interface design

Good interface design is *very* important - particularly for Argupedia whose users are members of the public debating on a platform guided by insights from AI argumentation research. In this sense, Argupedia has to perform an accessibility 'leap' to ensure members of the public are not frustrated or confused by the platform.

Wider benefits of good interface design includes maximisation of productivity and reducing users cognitive load spent on decision making [55]. Ultimately, this amounts to significant gains in user performance [47]. Cope and Uliano found that one graphical window redesigned to be more effective could save a company $20,000 in the first year of use [14]. Therefore, with respect

1. Visibility of system status
The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

2. Match between system and the real world
The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

3. User control and freedom
Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

4. Consistency and standards
Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

5. Error prevention
Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

6. Recognition rather than recall
Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

7. Flexibility and efficiency of use
Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

8. Aesthetic and minimalist design
Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

9. Help users recognize, diagnose, and recover from errors
Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

10. Help and documentation
Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Figure 4.1: Nielsen's landmark ten heuristics for designing user-friendly interfaces [53].

to Argupedia - a well designed interface will ensure users loyally return to enjoy, contribute and learn on the platform.

Oppermann provides an extensive analysis of further benefits from effective interface design accessible here: [55]. However, Karat et al., capture some of the nuances of good interface design in particular the notion that the success of the design is by providing affordances for the user to accomplish tasks and goals. In this sense, users want both personalisation, the ability to control, change platform and not be restrained by the platform [37]. In the next section, I will now discuss literature concerning the methodologies most appropriate for ensuring Argupedia is a well designed interface.

4.2 Design thinking process

Before expanding upon the design process prototyping and methodology in chapter 6, it is important to evaluate the academic literature concerning the design process. Argupedia will be developed via the five stage design process to best ensure all processes for designing a user-centred and effective platform - see figure 4.2 [15] have been accomplished. Advocated by experts in the field of HCI and design such as IDEO is following the five stage design thinking process [15, 27, 28].

The first stage, empathy - as noted by Helighen et al., focuses on gaining an empathic understanding of users of Argupedia and their respective needs [32]. Following this, is the define stage - in which the information gathered will be synthesised into a set of clear system requirements for the Argupedia platform [45]. After having established requirements, as noted by Reinig high quality ideation can begin for the platform - its interface, ergonomic principles and functionality [64]. The fourth stage of the design process will focus on building and testing prototypes. Before the final stage of Argupedia's design process is UX testing the developed platform.

A potential objection raised might be that the design process is too long and overly taxing - improperly misusing time and resources. However, in response to the objection - Argupedia should adhere to the upmost professional standards for design [27]. Indeed, Hales et al., systematically found the design process to improve design quality and all engineering activities of firms they analysed [28]. Additionally, as argued by Haik et al., no matter how good the sales, manufacturing, production etc., if the product is poorly designed it will ultimately always fail [27]. Having discussed the design thinking process more broadly, in the following section, I will analyse some of the key literature concerning user experience (UX) testing plus collaborative design and its importance for ensuring Argupedia is a successfully designed product.

4.3 User experience testing

Usability testing is a technique used in user-centered interaction design to evaluate a product by testing it on users [34]. Within the academic literature, UX testing is a well debated engineering tool - as noted by King et al., multiple versions of UX testing and experimentation have been designed and analysed to try and establish the most effective methodology [40]. Cook et al.'s book perhaps offers the most definitive and extensive analysis of experimentation and UX testing - explicating the difficulty of definitively proving causal inferences [13].

Despite this difficulty, UX testing of Argupedia must be performed to complete the design process and gain critical feedback on the success of the prototype [27]. Consequently, out of the potential options for UX testing - A/B testing is widely considered the most effective approach [34, 40]. As argued by both King, Nielsen and Kohavi, critical advantages of A/B UX testing is that it enables a very clear measure of performance differences and data on the user interactions instrumented and compared [34, 40, 42]. On the basis of these insights, it is important for Argupedia to be A/B tested versus a leading market competitor.

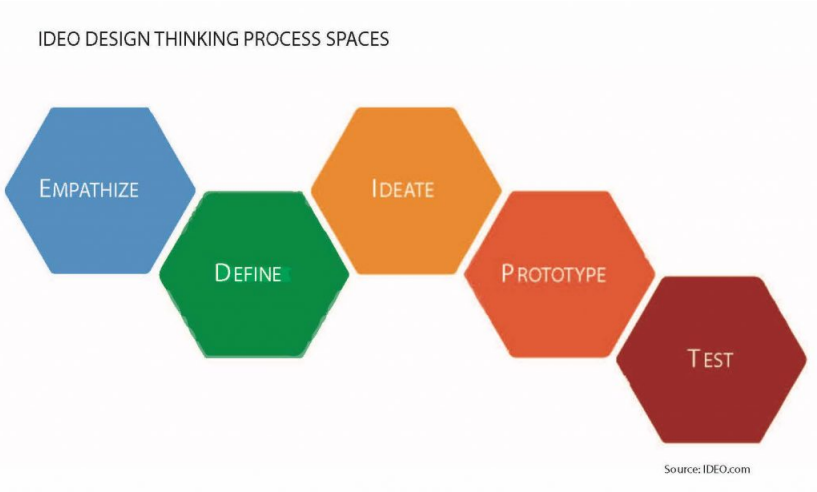


Figure 4.2: A diagram of all five stages of the design process pioneered at firms such as IDEO [21, 27].

4.3.1 Collaborative design

In developing a novel argumentation platform in an area with little precedent, there is a large risk that Argupedia becomes overly idiosyncratic - tied to the tastes, practices and accessibility limitations of myself as the researcher [43]. As argued by Kvan and Kleinsmann this problem is tackled by utilising collaboration and incorporating user feedback at the earliest possible stage [41, 43]. Consequently feedback will be incorporated at the earliest possible stage with friends during prototyping before eventual formal UX testing of Argupedia with participants.

Chapter 5

Rival argumentation platforms

This chapter focuses on competitor analysis of other public argumentation technologies for the purpose of improving the Argupedia prototype. Indeed, it is critical to identify and evaluate rival products and services - to look to improve upon their shortcomings and developing a competitive product. Additionally, this analysis provides contextual information and helps deduce Argupedia's unique selling point (USP).

Firstly, my analysis centres on analysing three market leading argumentation and debate platforms - their respective strengths and weaknesses. Following this, I then emphasize the role of Argupedia in mitigating these shortcomings and the advantages of Argupedia as a novel argument technology.

5.1 Kialo

Kialo is a novel peer production system focused on pro/con debate construction [6, 38]. Moderators also edit and refactor debates as they grow - playing an important role in cultivating and maintaining debates [6, 38, 80].

Advantages of Kialo include that it has an extensive amount of contributions at almost 2.5 million [38]. Furthermore, participants can debate any topic including: net neutrality, ad blockers, AI etc. [6, 38]. Lastly, Kialo does directly educate users by offering documentation on the theory and practice of argumentation [6].

Ultimately, there are two fundamental weaknesses of Kialo. Firstly, there is little insight for users on the current status of the debate - preventing users from understanding which arguments are winning or more preferred [6]. Therefore, users are not able to easily infer which arguments are best from debates.

The second additional prominent issue raised by research from Beck and colleagues is that moderators frequently conflict each other whilst managing debates - undermining collaboration and driving attrition [6]. Even more concerning is that Beck et al. find that the moderators attrition is typically derived from adversarial beliefs and values - moderators are thus not neutral [6]. Therefore, Kialo is not a suitably equitable platform because potentially biased moderators make key decisions about every debate. In summary, Beck et al. conclude that only Kialo will improve with more constructive mediation [6]. Having analysed Kialo, now I will consider another rival, namely Debate.org.

5.2 Debate.org

Debate.org is an online debating community that attempts to establish the reputations of its debaters democratically [18, 19]. The life cycle of a debate consists of four stages - Challenge, Debating, Voting and Post Voting [18, 19]. The winner of each debate is determined democratically through voting - every member of the community has the option to vote for the arguments they agree with [18, 19].

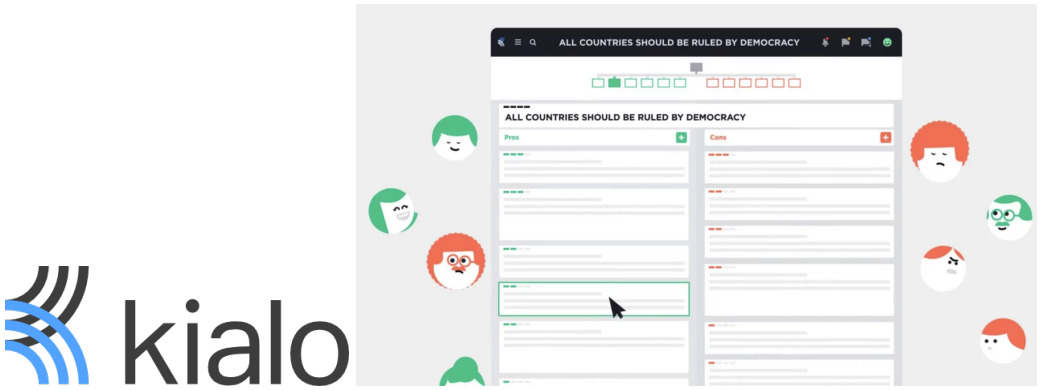


Figure 5.1: Kialo is a fast emerging online debate platform.

Figure 5.2: Two more of Argupedia's rivals - Debate.org and Reddit.

Advantages of Debate.org are that much like Kialo a wide pantheon of debate topics are encouraged [19]. Furthermore, it is the largest online debate platform globally [19]. Additionally, its precedence for democracy suggests that it will enable a wide range of debate from many sources and not support any forms of totalitarianism [18]. Research by Luu et al., has even found that users debating skills have markedly improved using the platform over time too [44].

However, empirical evaluations of Debate.org have raised serious concerns and limitations [19, 46]. Daniels et al., have raised several serious concerns - the most prominent being that the Debate.org is predominantly Christian from the United States (US) resulting in skewed voting and unmitigated cocooning of debates and values [18]. Additionally, Luu et al., concerningly find that debaters on the platform can be outright threatening using abusive language [46]. This disconcerting problem is certainly compounded when voting is transparent and votes cast are tied to accounts [18]. Following this analysis of Debate.org, now I will analyse another rival platform for Argupedia - Reddit.

5.3 Reddit

Reddit is a social news aggregation and discussion website. Registered members submit content to the website which are voted up or down by fellow members [63]. Posts are organised into user-created boards called "communities" or "subreddits" which cover a variety of topics [63].

Although not specifically a debating platform, advantages of Reddit include its popularity as the 18th most popular website in the US - resulting in a divergent range of opinions and users [63]. Furthermore, the potential for subreddits enables more niche topics of debate [63]. Much like Argupedia, Reddit threads have been successfully converted into argumentation graphs - not only this but Pazienza and colleagues weighted the attacks and support relations

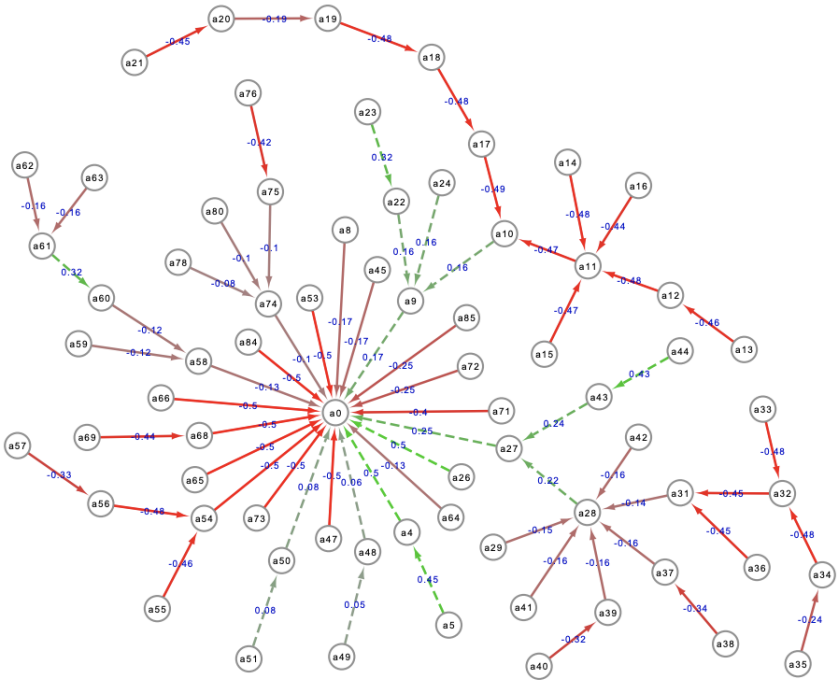


Fig. 3. BWAF representation for the considered Reddit Thread.

Figure 5.3: Pazienza and colleagues successfully employed machine learning techniques and bipolar weighted argument frameworks to model a debate thread from Reddit and produce a single argument graph similar to what Argupedia will produce automatically [56]. Ultimately, their approach took much time whilst Argupedia's users can build sophisticated debate graphs with comparable relative ease.

using machine learning techniques - see figure 5.3 [56].

Disadvantages of Reddit have been raised by Morini et al., here they note of the alarming emergence in Echo Chambers on Reddit - in particular they focused on Reddit being used to forge polarising pro-Trump and anti-Trump echo chambers [50]. Albota's studies of Reddit has noted that often conflict situations can result in unpleasant contradictory discussions, with the platform poorly moderated - high use of obscenities and ultimately a very limited and disjointed debate [1, 65]. Having discussed all three rivals, now I will provide an overall assessment versus Argupedia.

5.4 Assessment versus Argupedia

Similarities between all three platforms and Argupedia include that all the platforms enable debate on any topics of the users choosing. Additionally, much like Kialo, Argupedia also seeks to educate its users on debate and argumentation. Furthermore, Argupedia hopes for its debaters to improve at argumentation over time like users have been empirically found to improve on Debate.org.

Argupedia contrasts strongly with all three platforms in different ways. Initially, Argupedia differs from Kialo as it provides a running status of the debate using Dung's labelling algorithms and it does not at this present moment involve any form of moderation. Secondly, Argupedia differs from Debate.org as it enables anyone to contribute to a live debate rather than having two argue head to head. Furthermore, Argupedia has an impartial labelling algorithm to guide which arguments are most preferred rather than using a debating system with moderators.

Lastly, Argupedia differs from Reddit as it is a specific argumentation platform and educates its users on how to debate and argue. Indeed, unlike disjointed discourses on Reddit - Argupedia enshrines correct argumentative discourse through argumentation schemes and critical questions rather than letting users argue in a potentially disjointed and incoherent manner.

5.4.1 Argupedia USP

In light of the engineering design process considered in the previous chapter 4, this analysis of three competitor devices let me clearly and easily define Argupedia's USP. It is clear that Argupedia is unique because it is the only GUI application to utilise Dung argumentation graphs, educate users with formal argumentation schemes plus critical questions and provided users with a contextual understanding of the most preferred arguments according to Dung labelling algorithms. Additionally, Argupedia is the only application to offer an impartial overview of the status of the debate by using algorithms rather than human moderators and teach its users with argumentation schemes.

Unlike the polarisation, adversarial and disjointed nature of debate observed on other rival platforms. Argupedia guides user to structure arguments in a rational way using the template argument schemes and to direct their counter-arguments appropriately using critical questions. In addition, Argupedia offers an innovative graphical interface inspired by argumentation graphs which users can utilise to view how arguments relate to each-other in state space. Ultimately, debates from Argupedia serve a much more constructive purpose than rival platforms.

Chapter 6

Hypotheses & GUI prototyping

The focus of this chapter is twofold. Firstly, presenting the hypotheses to be used in the UX testing and early delivery of the Argupedia application. Secondly, prototyping for Argupedia with its system architecture and user stories. The focus of this project was on aesthetic outcomes, demonstrable interaction and thorough UX testing for future research. The lack of pre-existing research and platforms meant that a clear empirical and user-centric approach was most appropriate.

Throughout development of Argupedia for its pilot UX study - I incorporated an agile development cycle resulting in fast development of multiple prototypes and incorporation of feedback whenever possible. Due to the consequences of the Coronavirus pandemic and pioneering nature of the project I proceeded informally and opportunistically, utilising friends, family colleagues and even self-study whenever possible.

6.1 Hypotheses for UX testing

For the pilot UX testing, I formulated two central hypotheses for the project. Both of my hypotheses centre on qualitative evaluation and data collection from user feedback collected using an Attrakdif survey and interviews.

Hypothesis 1 (H1): *Participants will find the Argupedia application more user-friendly and engaging than the control Debate.org.*

Hypothesis 2 (H2): *Participants will learn more about the philosophies of argumentation, logic and debate from having used Argupedia.*

The required qualitative insights will come from interviews and questionnaire data. From this data I can evaluate hypothesis 1 to see if Argupedia has been easy to use requiring low cognitive output and engaging versus a rival device namely the current most popular online debate platform, Debate.org in the form of A/B UX pilot testing. In contrast, for hypothesis 2 I can determine if Argupedia has been an effective teaching experience based on a self-reported qualitative feedback from participants i.e. which platform has enabled users to engage and learn most about argumentation and debate.

6.2 Initial design reflection

The following section discusses the pros and cons of design thoughts taken during initial research for Argupedia. Some ideas proved fruitful and were implemented in the final design. Since Argupedia is a novel implementation it felt appropriate to reflect on the ideation and prototyping of the design. Furthermore, for this part of the engineering design process - I drew inspiration from a variety of iPad sketches exhibited by Professor Modgil and looked to create more formalised prototypes using an online software called Wireframe.cc.

Some of the methods were adopted whilst others were unsuccessful initial ideas. Therefore, advice for future computer science students embarking on similar projects is to get started early and not be deterred by initial lack of success. The initial core design idea was strongly shaped by advice from Professor Modgil - to build a web application which lets users define arguments using schemes, sketches argumentation graphs and add labellings.

6.2.1 Argupedia GUI

Prior to the first meeting, Professor Modgil using two videos broke down his vision of how the Argupedia application should work using two videos. Notably, he emphasised that we should take ownership of the project and add our own independent vision.

From this, I tried to break down some of the points he made and transfer some of the sketches Professor Modgil presented on an iPad into more formalised sketches. This process was very successful as it enabled me to consider alternative designs and experiment before settling on a definitive outcome. Initially, I envisaged for the Argupedia platform to be totally different from competitor devices and a new and dynamic interaction experience.

As denoted by figure 6.1 - the first step was to dynamically build a form dependent on argument scheme selected by the user. The user would then input their respective argument.

The second step as shown by figure 6.2 was to enable users to generate counter-arguments. Firstly, presenting them dynamically with critical questions related to the initial argument. Then regenerating the initial argument scheme to enable them to input a counter-argument. Here I even toyed with the idea of letting users click the arguments they wanted to counter-argue against.

As demonstrated by figure 6.3 the third step focused upon creating dynamic Dung argumentation graphs of the exchanges. Multiple different graph designs were considered - as well as the display of different information about the arguments within the nodes. Also the potential for hovering functionality on the nodes was considered as a potentially exciting and dynamic application.

Finally as shown in figures 6.4, 6.5 and 6.6, users would need some sort of button or switch to signal for a labelling algorithm to be applied and verify the winning arguments according to the different labellings. Different colours such as a traffic light system (i.e. IN=Green, YELLOW=UNDEC and Red=OUT) was theorized as a potentially visually stimulating option for the argument labelling algorithm. Lastly, I will now consider potential system landscapes to operate Argupedia successfully.

6.3 Prototypical system landscape

From conception, I decided to incorporate a dynamic, usable and aesthetically pleasing front-end with the labelling algorithms and database operating in the back-end - please see the architecture diagram in figure 6.7 for the prototype.

The database was envisaged to be document-oriented format - providing the potential for easy CRUD operations, high performance, speed, flexibility for changes due to its schema-less structure with no foreign keys and due to my personal familiarity.

A state diagram in figure 6.8 was developed for the front-end to enable me to focus my development on the different processes and potential use cases.

6.4 User stories

To aid development I devised three short user stories to detail different interactions that a user may have with Argupedia. Devising user stories was an invaluable tool for understanding how users would utilise Argupedia and to demonstrate the purposes of the platform. Furthermore, user stories helped orient the platform toward the profile of users the device was best suited too.

6.4.1 User story 1

Mika is a PhD student based at the University of Dundee. He has specialised in AI argumentation as a part of his degrees. He is keen to use members of the public to provide data for an argumentation AI he is developing.

6.4.2 User story 2

Sara is a middle aged mum, she is a former politics student and is interested in philosophy. She does not have the time to return to full-time education but wishes to be intellectually stimulated and to learn more about argumentation and debate.

6.4.3 User story 3

Dan is a school student on the debate team. He is keen to improve his ability via debating effectively online. He has aspirations to debate for the Great Britain team.

6.4.4 Conclusions from user stories

Concluding this chapter, I used agile practices to understand the potential hierarchy of priorities from users as well as develop a clear profile of the user base. From the user stories it is clear a wide age range of potential clients would use Argupedia.

All three users emphasised a need for further education or looked to learn from the platform. User story 1 and 3 are enrolled in full time education and sought Argupedia for academic purposes, whilst user story 2 sought learning. All three user stories emphasised a definitive interest in argumentation and debate at different levels. User story 2 at a less academic level versus the elite level of user stories 1 and 3. Ultimately, Argupedia will have to be configured to be usable for different levels of argumentation from elite users to more novice users and different age ranges. Additionally, users of Argupedia will most probably be interested in learning from the platform about argumentation and debate.

User story 2 and 3 explicitly demand to improve and have performance aspirations on the basis of utilising Argupedia - they both wish to improve their abilities through Argupedia. In contrast, user story 1 already has confidence in his abilities and is seeking data from the platform. All three users have received full-time education in some capacity in the past and are well-educated. Admittedly, user story 2 may have only studied up to school level. Nonetheless, Argupedia could still appeal to those without a full-time background in formal education.

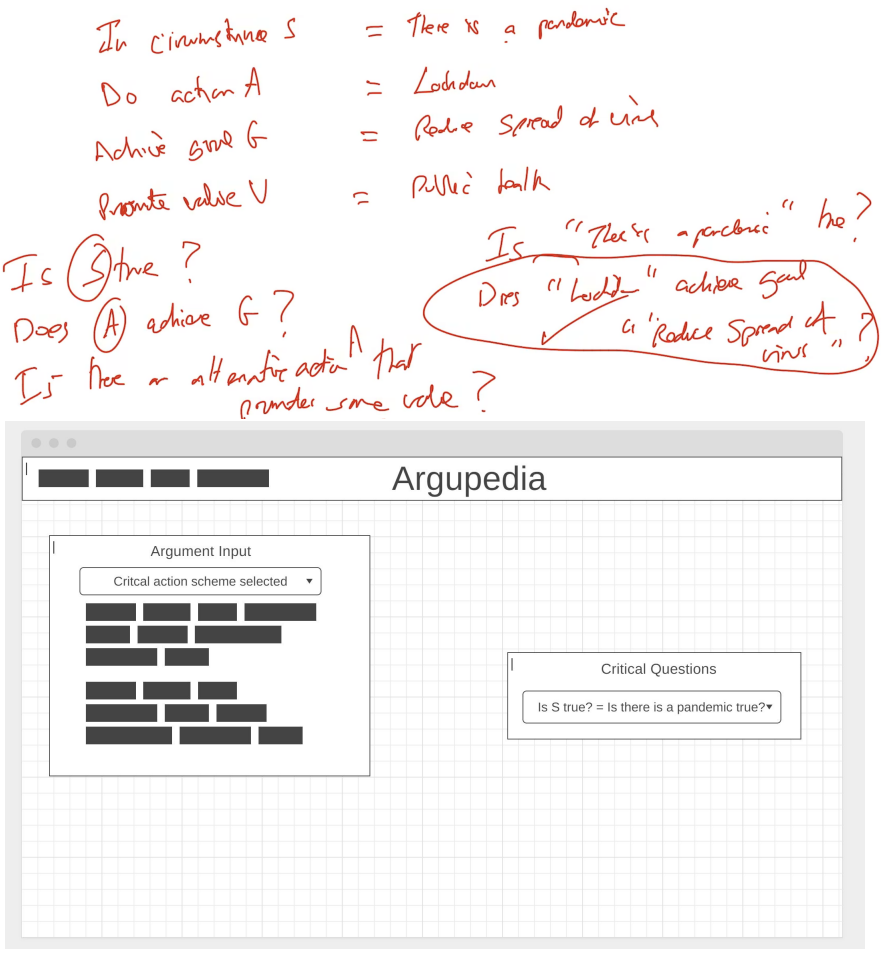


Figure 6.1: Envisaged argument scheme form drawn by Professor Modgil in red (above) versus a Wireframe.cc prototype of Argupedia (below) where users input the selected scheme via a form and select critical questions via a picker.

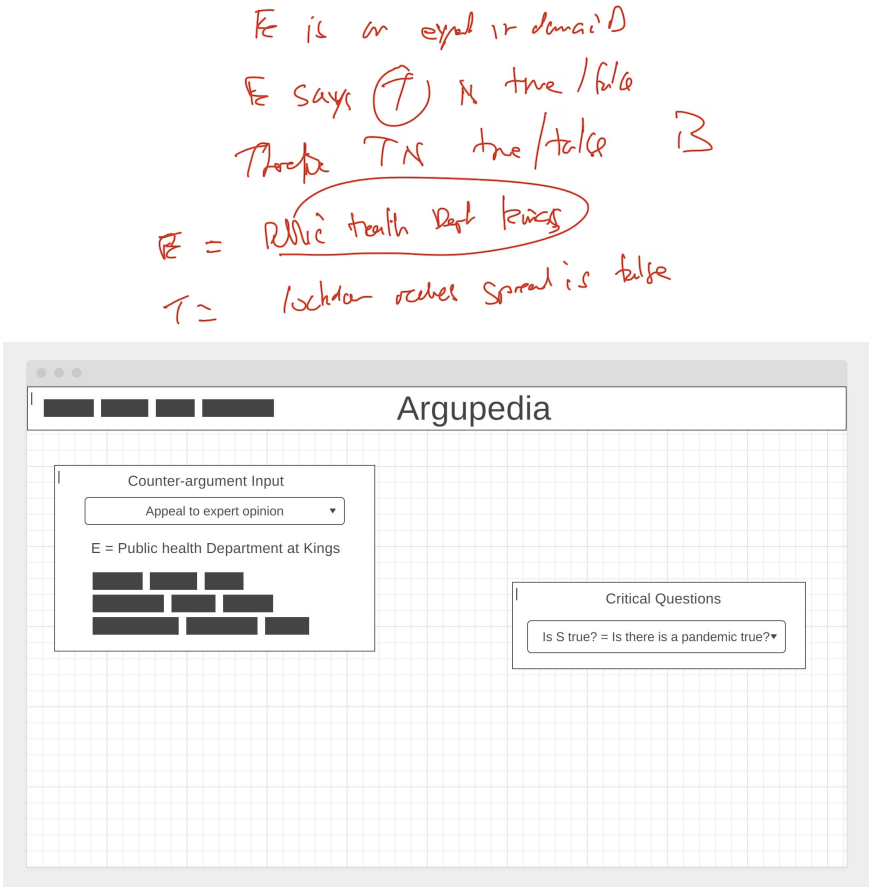


Figure 6.2: Sketched argument scheme form from Professor Modgil (above) and a prototype (below) demonstrating variables of an argument being filled in the form.

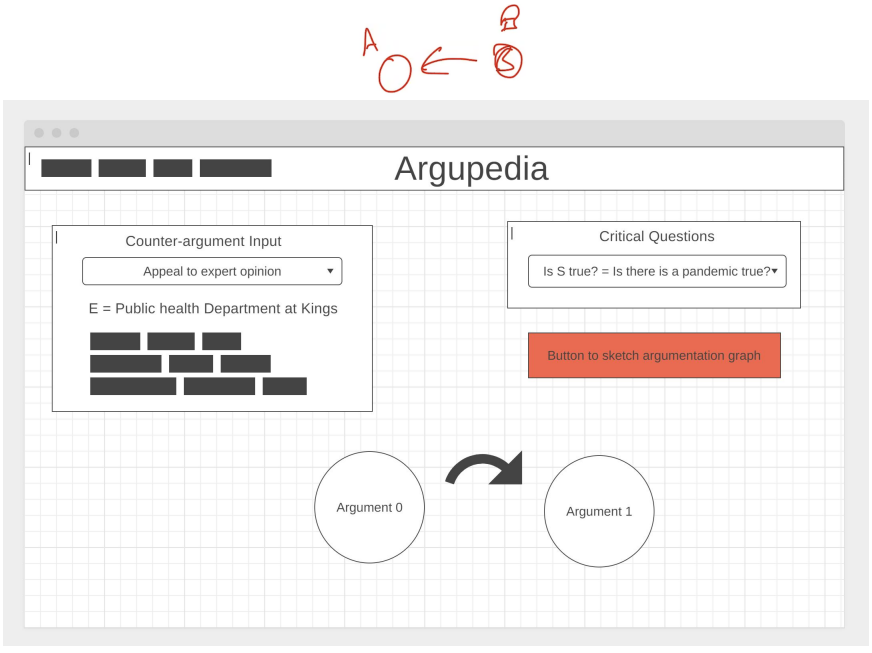


Figure 6.3: Sketch demonstrating argument graphs (above) and a prototype of an envisaged graph button to create argument graphs within the Argupedia application (below).

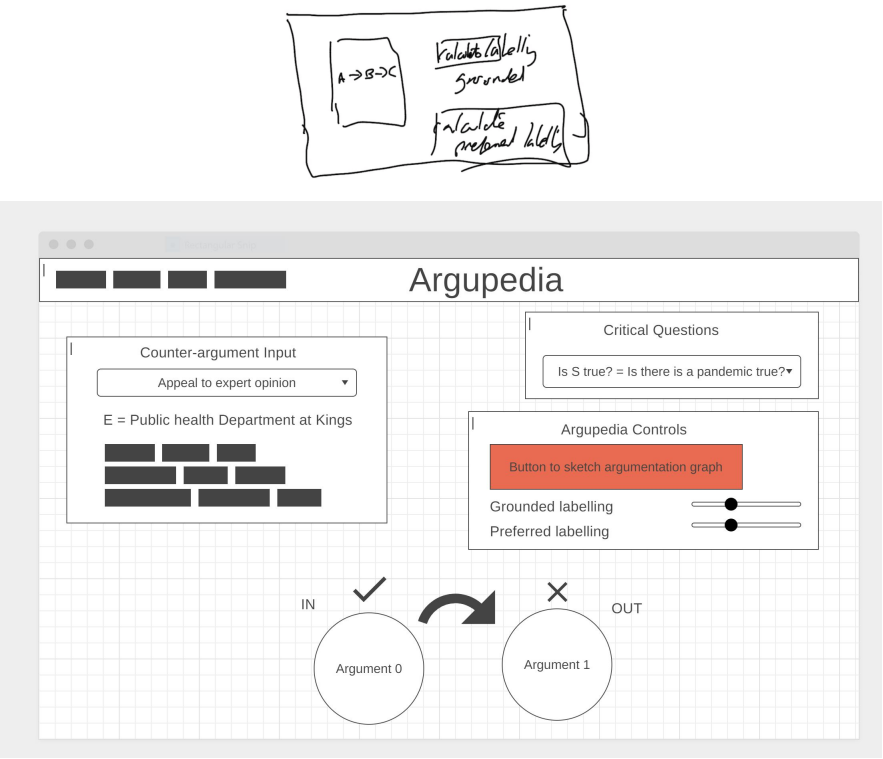


Figure 6.4: Professor Modgil sketch of application in black (above) with prototype built (below) demonstrating switches to add to the argument graph labelling algorithms.

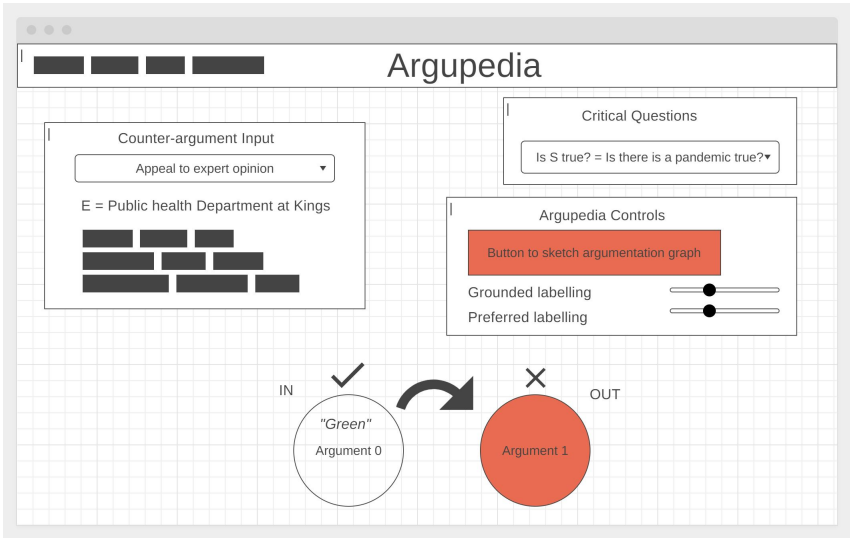


Figure 6.5: Labelling algorithm expressively denoting which arguments are IN versus OUT on the argument graph (above and below) through ticks and effective colouring.

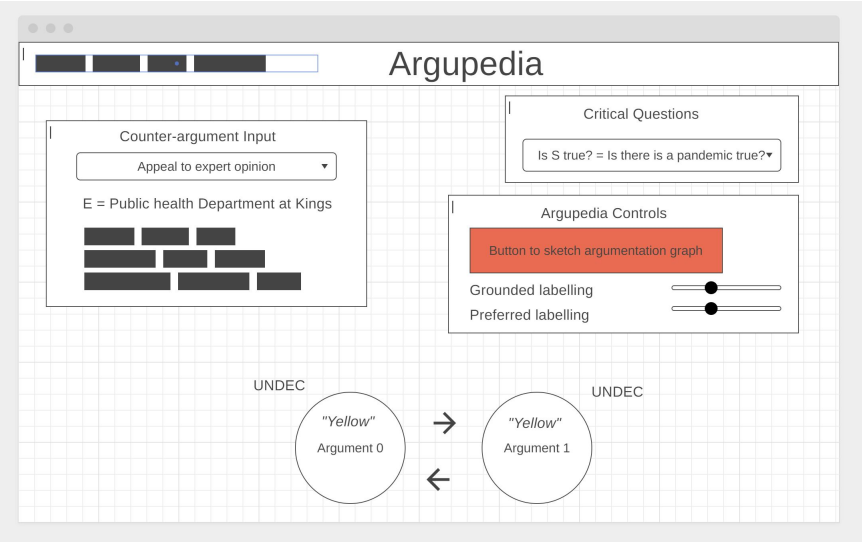


Figure 6.6: The potential for arguments to be IN, OUT and UNDEC (above). Arguments that are UNDEC expressively denoted in "yellow" (below).

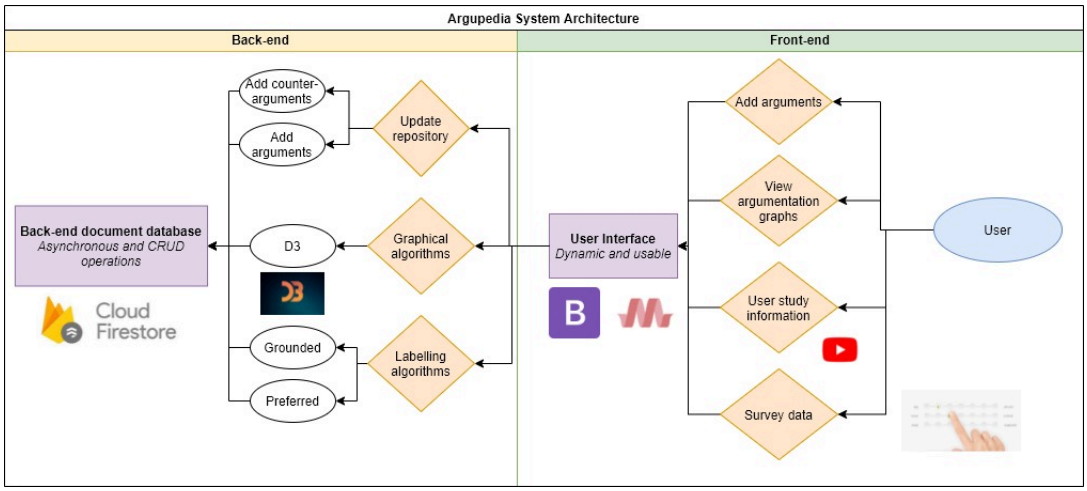


Figure 6.7: An envisaged architecture diagram for Argupedia to operate.

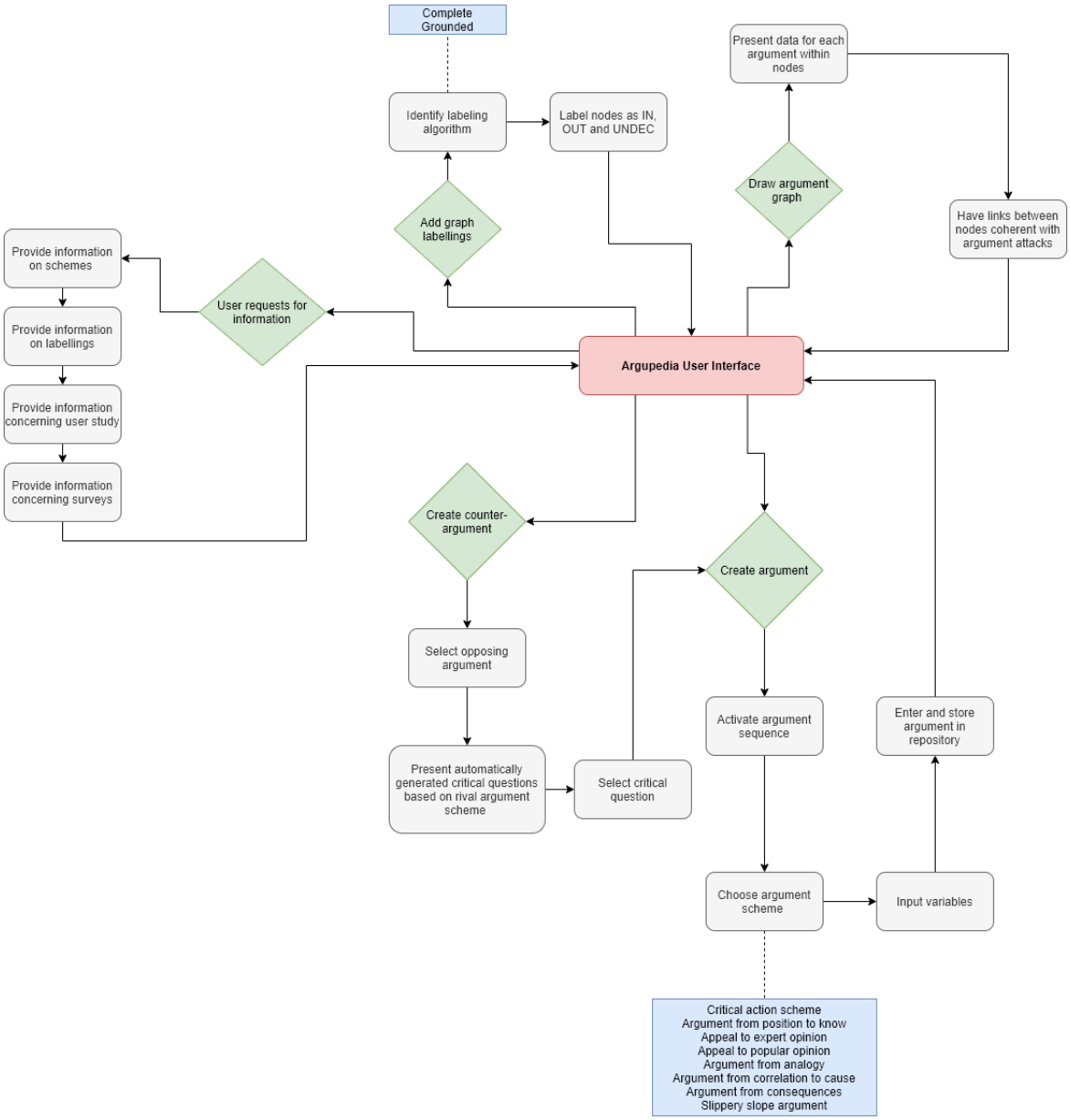


Figure 6.8: An Argupedia state diagram of all the possible stages and changes of the interface for the user.

Chapter 7

Specification & requirements

This chapter initially lists the requirements of the Argupedia project - firstly, the base minimum level of requirements and secondly, the more advanced requirement extensions for the project. Afterwards this chapter will then detail a focused specification to meet all the requirements.

The two-tiered nature of the Argupedia requirements worked well - enabling there to be initial focus on a more simplistic Argupedia prototype. Before extending the prototype in directions that were exciting and individual to our research interests. Also the specification can help shape future directions of research for the Argupedia project.

7.1 Minimum requirements

1. Create an updatable back-end database for Argupedia's internet repository for arguments, debate and opinion.
2. Data entry from users on Argupedia has to conform to an argument scheme conforming to principled theories of argumentation.
3. Within the Argupedia GUI users must successfully initiate initial lines of argument.
4. Within the Argupedia GUI users must easily initiate counter-arguments to attack rival arguments as well as receive critical questions.
5. The arguments must be posted in Argupedia in a graphical way, inspired by Dung argumentation graphs that can show how debates evolve and arguments relate to each other over time.
6. Users can then initiate an evaluation of the arguments within Argupedia with labelling algorithms of the argumentation graphs.
7. Users can request on Argupedia the grounded labelling criteria.
8. Users can request on Argupedia the preferred labelling criteria.

7.2 Extended requirements

9. Add the complete labelling criteria.
10. Argupedia is a usable interface - equipped with a well styled and learn-able GUI.
11. Within the Argupedia GUI, there are multiple different extended varieties of argument scheme.
12. Effectively host Argupedia on a live website server for members of the public to use.

13. Argupedia receives initial user testing and feedback with 3-5 participants.
14. An insightful YouTube video explains to participants how to effectively use Argupedia.
15. Argupedia enables users to use a voting algorithm to vote for their preferential arguments.
16. Argupedia provides users with the ability to have supporting arguments not just counter-attacking arguments.
17. Argupedia has a user login.
18. The design of a new Argupedia web-page back-end to enable easy uploading of further schemes.

7.3 Minimum specification

Requirements	1 - *minimum*
Priority	- High
Purpose	- Configure an updatable back-end database to serve as a repository for user inputted data through the Argupedia application.
Operations	- Decide upon a database format for user argumentation data input. - Trial different types of database. - Develop JavaScript code to asynchronously update the database.
Output	- Storage of users arguments and successful creation of a repository for future arguments. - Successful storage of data for users arguments and the links between arguments. - Enable database to be queried and called on the front-end.

Requirements	2 - *minimum*
Priority	- High
Purpose	- Ensure that users argument input conforms to principled theories of argumentation.
Operations	- Develop dynamic JavaScript forms on the front end for user argumentation input. - Ensure forms successfully submit user data to the database back-end.
Output	- Users will be educated and able to submit their arguments in accordance with argument schemes. - All arguments submitted will conform to principled theories of argumentation.

Requirements	3 and 4 - *minimum*
Priority	- High
Purpose	- Enable users to initiate initial lines of argument and counter-arguments against pre-existing arguments.
Operations	- Augment JavaScript code for user-input on the front-end to let users specify initial argument versus counter-argument. - Query database to provide critical questions for users when counter-arguing. - Update database on the basis of input intial arguments and counter-arguments.
Output	- Users can state initial lines of argument using argument schemes. - Users can challenge pre-existing lines of argument having been provided with critical questions.

Requirements	5 - *minimum*
Priority	- High
Purpose	- Enable users to visualise arguments and their evolution graphically in accordance with Dung's argumentation graphs.
Operations	- Develop a directed graph of user-inputted arguments using libraries. - Have the graphical component update asynchronously on the basis of changes to the underlying database. - Have the graph successfully display initial lines of argument and counter-arguments.
Output	- Users can graphically visualise arguments, their interelation and the debate change over time.

Requirements	6, 7 and 8 - *minimum*
Priority	- High
Purpose	- Enable users to apply labelling algorithms to the predefined argumentation graph. - Provide users with a contextual understanding of the arguments at present most preferred and in effect winning the debate.
Operations	- Dynamically adapt the front-end to enable users to signal when they wish to label the graph. - Configure Dung et al.'s labelling algorithms into JavaScript code. - Label the graph on the basis of the requested labelling. - Deduce most preferred arguments. - Redraw the graph on the basis of the labelling.
Output	- Users can successfully determine which arguments are most preferred in accordance with dynamic labelling algorithms. - The graph is redrawn with requisite labelling.

7.4 Extended specification

Requirements	9 - *extended*
Priority	- High
Purpose	- Add a further labelling algorithm namely the preferred.
Operations	- Dynamically adapt the front-end to enable users to signal when they wish to label the graph. - Configure Dung et al.'s labelling algorithms into JavaScript code. - Label the graph on the basis of the requested labelling. - Deduce most preferred arguments. - Redraw the graph on the basis of the labelling.
Output	- Users can successfully determine which arguments are most preferred in accordance with dynamic labelling algorithms. - The graph is redrawn with requisite labelling.

Requirements	10 - *extended*
Priority	- Medium
Purpose	- Adapt the GUI to be usable, learnable and not a frustrating GUI experience for users to debate. - Use styling to improve user experience on the Argupedia GUI.
Operations	- Adapt code to have the requisite forms for argument schemes dynamically appear. - Have a visually appealling and navigatable argumentation graph with appropriate use of colour. - Make the front-end minimalist, clear and coloured logically. - Appropriately use buttons to make the GUI clear and usable.
Output	- Even novices can grapple and attempt to use Argupedia - learning more about debate and argumentation. - Have an accessible, positive and effective user experience.

Requirements	11 - *extended*
Priority	- Medium to high
Purpose	- Enable users to not be limited whilst arguing by providing them with several different various argument schemes.
Operations	- Identify the schemes and critical questions and convert them into dynamic code. - Adapt the JavaScript code to create critical questions for the new argument schemes dynamically. - Adapt the front-end to dynamically change the input forms to the user requested argument scheme. - Adapt the back-end code to store different types of argument using different schemes. - Configure the argumentation graph to sketch different types of arguments. - Provide explanation for the different types of argument scheme.
Output	- Users can utilise several variations of argument scheme to argue successfully. - Users are presented with the requisite critical questions for each scheme.

Requirements	12, 13 and 14 - *extended*
Priority	- High
Purpose	- Host Argupedia on a webpage with an educational YouTube video. - Have 3-5 users test the Argupedia prototype. - Gain clear qualitative data in the form of UX testing on Argupedia - its current effectiveness and feedback on future pathways for research.
Operations	- Host the webpage prototype publicly on a server. - Develop a model of user experimentation to collect data effectively. - Adapt the front-end providing clear instructions for user testing. - Configure the front-end with a YouTube video tutorial for the Argupedia prototype. - Get participants for UX testing. - Develop questionnaires and seek interviews to collect UX data from test participants.
Output	- Argupedia is effectively user tested in a pilot study. - Actionable feedback and data is gathered for future research. - The strengths and weaknesses of Argupedia are immediately identified.

Requirements	15 - *extended*
Priority	- Medium to low
Purpose	- Employ a voting algorithm to enable users to vote for their most preferred arguments.
Operations	- Adapt the front-end GUI to register votes for arguments. - Adapt the back-end to store votes for arguments. - Redraw the graph on the basis of votes changing the links between arguments.
Output	- Enable Argupedia to become more democratic and innovative via the ability for users to vote on their most preferred arguments.

Requirements	16 - *extended*
Priority	- Medium to low
Purpose	- Have users login and add arguments with an account.
Operations	- Adapt the back-end to store user data for accounts. - Using the login, collect data for the back-end on users, arguments initiated and their interaction with Argupedia.
Output	- Provide users with an account and login for Argupedia.

Requirements	17 - *extended*
Priority	- Medium
Purpose	- Create an easily usable webpage to enable a layperson to add new argument schemes to Argupedia.
Operations	- Develop a new web application. - Have the web application add argument schemes to the database. - Configure Argupedia to work off the argument schemes stored in the database.
Output	- Argupedia can be extended with new argument schemes by a layperson with no working knowledge of code.

7.5 Discussion of specification

The specification was very successful in classifying minimum requirements outlined by Professor Modgil explicitly during meetings and to extend the specification with requirements for potential extended avenues of the project. The specification has been designed deliberately to offer up potential future pathways for research as not all requirements have been fully met. Nonetheless, from my perspective at the initial specification stage it was good to have bold aspirations and to add a significant extended section to the specification.

Chapter 8

Tools

8.1 Platforms

8.1.1 Operating systems

Development for Argupedia consisted on Windows 10 using a Linux subsystem and Mac OS mainly because I was able to use two laptops with both operating systems. Furthermore, all the major development frameworks I considered worked on either OS. In contrast, working on Linux in a virtual machine did not seem an option due to the performance demands of the project.

8.1.2 Firebase cloud firestore

Google's Firebase platform was used to develop Argupedia as a mobile and web application. In particular, the Firebase Cloud Firestore served as a free host of Argupedia's database and was primarily chosen as it had extensive documentation coupled with useful discussion forums to aid development. Importantly, the Cloud Firestore added a great deal of stability to Argupedia as it synchronizes the data in NoSQL format in real time thereby ensuring the application remained usable in the case of loss of internet connection. Furthermore, the flexibility, expressive querying and scalability serves as a fantastic bonus for future development of Argupedia.

8.1.3 YouTube

YouTube proved pivotal for hosting the tutorial video for Argupedia and generating HTML code to immediately and easily integrate with the Argupedia web application.

8.1.4 GitHub Pages

GitHub pages was a useful resource for hosting the live version of Argupedia directly from my repository. Not only, was GitHub pages free but all I ever had to do was edit the underlying code, push my changes and they were subsequently live. In the future, GitHub pages offers excellent scalability with the potential for custom URLs and several guides on 404 pages and sub-modules.

8.1.5 Wireframe.cc

Wireframe.cc is a fantastic free wire-framing app which provides a unique URL that you can bookmark and share. The app was critical for allowing me to create multiple fast prototypes of the Argupedia GUI which enabled me to consider a plethora of alternative designs.

8.2 Frameworks

8.2.1 Materialize

Google's Materialize responsive CSS framework was used significantly to style the Argupedia application front-end. Once again the extensive documentation for the platform proved very useful. Additionally, the well styled basic components, responsiveness and use of motion helped provide extensive meaning and familiarity for users to Argupedia's front end.

8.2.2 Bootstrap

Bootstrap was also used to help design and style the Argupedia application front-end. Bootstrap's consistent UI, JavaScript and JQuery plugins proved immensely useful. Furthermore, the significant documentation, my personal experience with the framework and ease of use meant that Bootstrap was a fantastic tool for developing the Argupedia front-end.

8.3 Libraries

8.3.1 D3

D3 is a crucial JavaScript library for manipulating documents based on data generated by the users within Argupedia. The latest library is easy to use as you can install with just a simple script call. D3 helped bring Argupedia data to life using just HTML, SVG and CSS and its monolithic framework seeks to provide every conceivable feature. Additionally, the documentation is thorough with a wide variety of visualisation collections, easy pre-existing collections and extended functionality. Looking to future development of Argupedia D3 even enables animations, interactivity and plots making it effective to use.

8.3.2 Dagre D3

Within the D3 JavaScript ecosystem a particular library namely Dagre was utilised to draw Argupedia's directed argumentation graphs. Dagre proved invaluable due to the availability of key forums and documentation. Dagre is a very new library and consequently some features such as labellings of nodes are not fully supported. Nonetheless the forum discussions were critical for development. Dagre enabled the drawing of detailed and high level acyclic graphs integrated with labelling algorithms.

8.3.3 Favicon

Favicons were used as a simple shortcut icon to make the Argupedia web-page more amenable and navigable - improving the Argupedia front-end interface.

8.4 Toolchain

8.4.1 Visual studio

Argupedia's GUI was written in HTML, Javascript and CSS within Visual Studio. The majority of the project was developed with JavaScript and JQuery using dynamic web development wherever possible. A number of useful plugins were adopted for the development of Argupedia. In particular, Live Server which enabled quick development with live browser reload.

8.4.2 Version control

I used Git for version control, hosting my repository with GitHub. This decision was based on my experience with the system, pro-license through Kings' College London and the availability of free hosting.

8.4.3 Zoom

Zoom was the primary video communication technology software used to meet and interview users who trialled the Argupedia prototype. The software was favoured due to user preference - users for testing felt most familiar and comfortable with the peer to peer platform.

Chapter 9

Methodology & implementation

Within this chapter I will focus on the development and implementation of Argupedia as well as a justification for the methodology. This development progressed in three stages. The first stage was prototyping discussed in the previous chapter 9. The second stage was implementation discussed in this chapter. The final stage focused on experimental design discussed in the next chapter 10.

Firstly, I will provide a broad system overview of the Argupedia project. Before secondly, going through the specification discussing completed facets of the project - I have subdivided the specification development into front-end and back-end development. Except where explicitly stated with comments every design element and implementation discussed in this chapter is my own work.

9.1 System overview

The software divides into three layers - a repository for arguments based on Walton's argumentation schemes, labelling algorithms and graphs developed off Dung et al.'s AI argumentation research and a dynamic interface and multimedia layer for ease of user experience. Elements of these layers are mingled in the discussion with a focus on novel aspects of development rather than commonplace aspects of implementation. Details of the tools used are discussed in chapter 8. However, the system is best understood via cataloguing the software into back-end development and front-end development.

9.2 Back-end development

9.2.1 Databases for an argument repository

Google Firebase's Cloud Firestore served as the optimum database for storing Argupedia's argumentation data. Asynchronous calls to the database were implementable and the documentation was excellent. Furthermore, the server-less and scalable architecture meant that time was not wasted on unnecessary tasks. Additionally, the latency of the database, offline synchronization and highly user-friendly database portal worked well during development.

Based on the graphical D3 library used, two Cloud Firestore collections and their enclosed documents were required to get the desirable front-end output. This meant two asynchronous calls to the database to get: initially, the arguments and their encapsulated data, followed by data concerning the arguments inter-relationship and the links between arguments i.e. the arguments that are attacking or counter-attacking.

Another asynchronous call would be need to be made to the database to support critical questions - once again, Firestore proved useful because it supported indexed queries and ACID transactions. This meant that I could make real-time database calls for the argument and its inputted variables to dynamically append critical questions with appropriate variables filled.

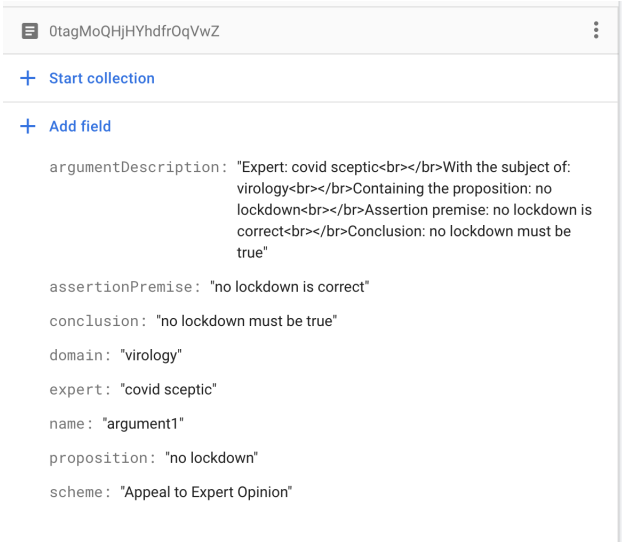


Figure 9.1: Data encapsulated in single argument submission including the argument name (ID), argument scheme, argument description and user provided variables.

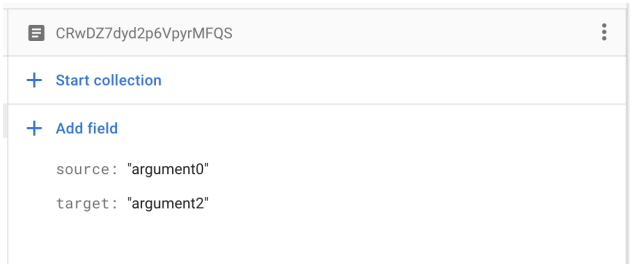


Figure 9.2: The links between arguments were stored in another Firestore collection separately and simply store links between argument IDs.

This feature also enabled an asynchronous call to the database to query the number of arguments within the collection - providing useful data for the labelling algorithms. During all testing the performance and latency of these calls were found to be excellent.

Lastly, I had to support users being able to submit their arguments to the repository. Here, I was initially unsure with what data to encapsulate in each argument submission for the user because of the differing argument schemes. However, Firestore was very extendable letting me submit as many variables as I liked for each users argument as pictured in figure 9.1. Consequently, after troubleshooting - for each argument submitted to the database was: the argument name to serve as front-end ID, the argument scheme, the full user argument and the variables submitted by the user for each argument. Notably, Firestore also dynamically created its own ID for each submission.

9.2.2 Development of labelling algorithms

For the labelling algorithm I needed a back-end call to get all the arguments and their links - see figure 9.2. Following this I was able to covert Dung et al., argumentation logic into JavaScript code. Initially, I divided arguments into arguments being attacked and arguments not being attacked - the arguments with no attackers automatically received an IN labelling.

Having added the labelling, I then developed code to label the remaining unlabelled arguments IN or OUT dependent on if they were attacked by IN arguments at all and if they had at least one attacker labelled as IN. Finally arguments became labelled UNDEC if they could

not be conclusively defined as IN or OUT. The algorithm is designed to run for the number of arguments within the graph and thus adapt given the differing sizes of the argumentation graph - this data was fortunately accessible thanks to Firestore's indexed queries.

9.3 Front-end development

9.3.1 Supporting Walton's argument schemes and critical questions

During the prototyping stage, my goal was to make the front-end as usable as possible for successful UX testing with lay-people totally unfamiliar with AI argumentation. Consequently, I decided upon a minimalist approach in which the users would only be presented with relevant choices, controls and options at any given time. This would enable the user to progress through the correct sequencing of creating a valid argument and not be overwhelmed with too many options and choices - resulting in an incorrect argument submission to the database.

Front-end frameworks Materialize and Bootstrap enabled me to implement modal popups which I could then manipulate with JavaScript code to append and remove the relevant forms, buttons and options - ensuring correct data input. Adding another level of difficulty, I correctly synthesised the JavaScript code so that in the process of adding a counter-argument - the relevant critical questions would also be able to be appropriately dynamically generated. Although this was difficult and often challenging - the end output was certainly worthwhile and very usable.

On the modal, for each argument scheme I had an appropriate text label to make it clear for users to input the correct argument scheme variable - ensuring users input the correct data to be stored in the back-end and to generate critical questions for future counter-attacking arguments. Finally, the code had to be developed to support all 8 argument schemes. Synthesising the code carefully - I managed to boil down adding a new argument scheme to around one hundred lines of JavaScript code - making it relatively painless to add a new scheme and appropriately configure the Argupedia front-end to generate argument scheme forms and dynamic critical questions.

9.3.2 Producing Dung argumentation graphs

A subset of the D3 library was used to produce Dung argumentation graphs called Dagre D3 - this library was used as it accurately produced the correct direct acyclic graphs required for Argupedia. Other benefits of the library include that it enabled the addition of useful functionality such as zooming and free movement around the visualisation of the graph. For the library functions to render the graph I had to present the nodes and links between nodes, which meant asynchronously calling and updating these from the back-end database.

In contrast, adding labelling algorithms was more challenging. Fortunately, the DagreD3 library provided a clear aesthetic and styling of nodes. On the front-end, users defined which labelling they required through explicit switches. For labellings, I had to first employ my labelling algorithm to categorise the argument nodes and then render the nodes with different labels and colours (fills) dependent on whether the argument was: IN i.e. green colour, OUT i.e. red colour or UNDEC i.e. yellow colour. In addition, explicit text dynamically appears below the graph to signify which arguments are IN under the grounded and preferred labellings - promoting increased accessibility. Weaknesses of the Dagre D3 library are that it does not yet support exterior labels on the argument nodes. Nonetheless it let me add all necessary functionality.

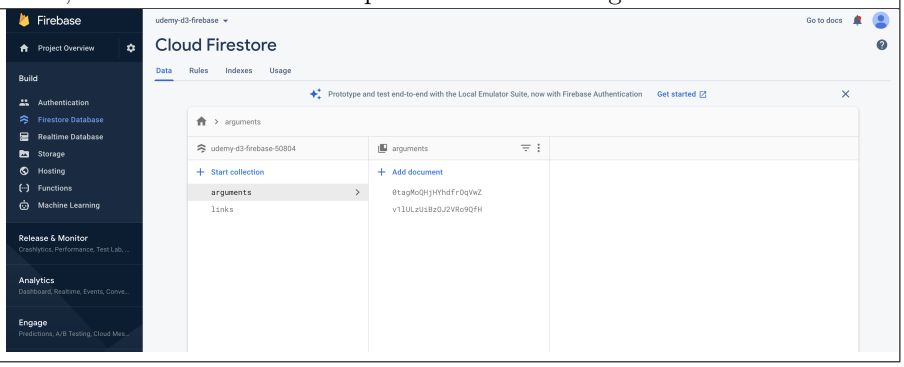
For counter-arguments with conflicting claims - on the modal I let users specify whether the counter-argument they were generating had conflicting claims. This was different to other approaches where Argupedia would perhaps recognise that the claims were conflicting on the basis of critical questions chosen or parsing the premises of either argument. Ultimately, I wanted to provide users with all the tools at their disposal and teach them the difference between counter-arguments with or without conflicting claims. Indeed, it would perhaps be confusing for end users to have Argupedia automatically generate arguments with conflicting

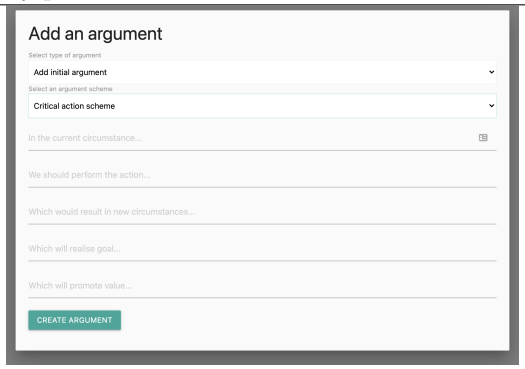
claims and not fully understand the causes. Nonetheless, Dagre D3 generated arguments with conflicting claims correctly with bidirectional arrows.

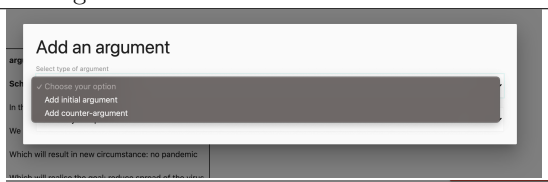
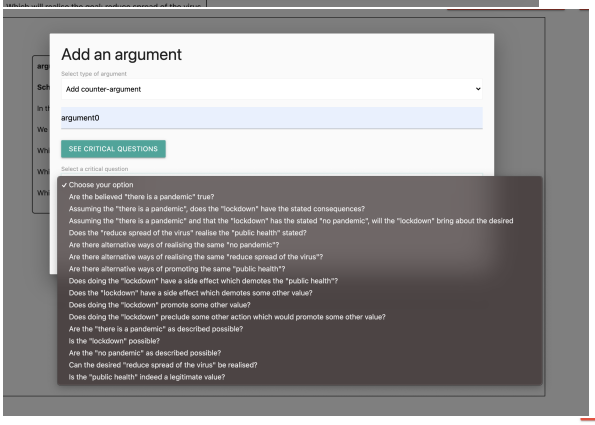
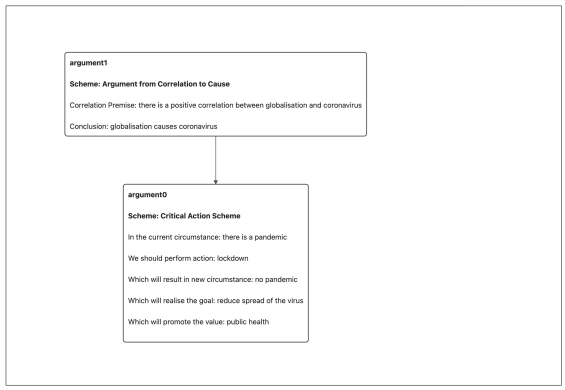
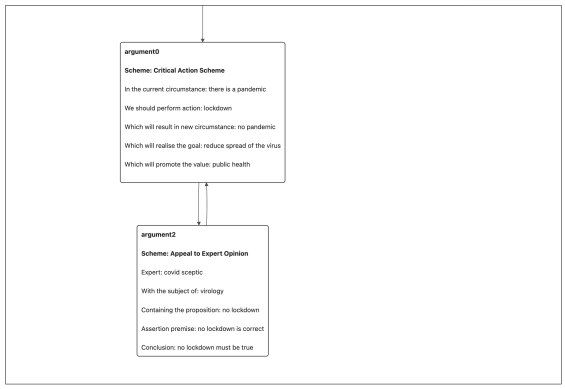
9.3.3 Development for UX testing

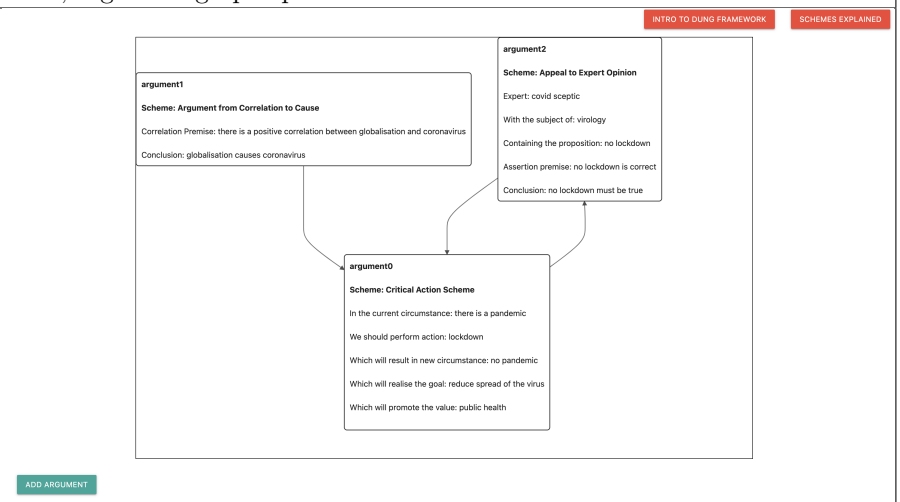
For Argupedia's A/B UX testing to be most successful and due to the constraints of the pandemic (it occurred over long distance) - meant adding to Argupedia's web interface to make independently testing the platform as easy as possible for participants. Consequently, I developed a homepage equipped with full instructions, embedded a YouTube video outlining precisely how to use Argupedia and included all necessary links for questionnaires and further information so that a participant could work through the study straightforwardly in sequential order.

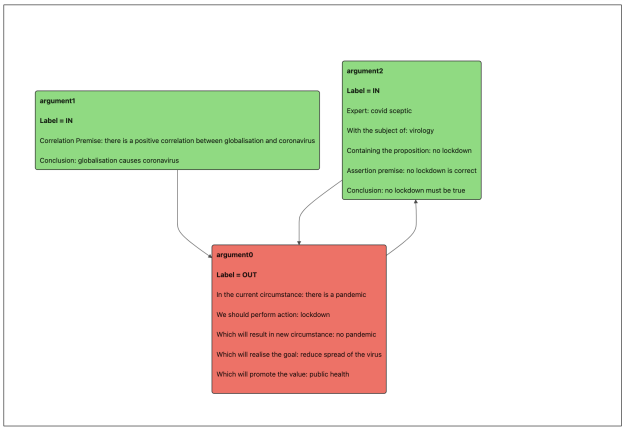
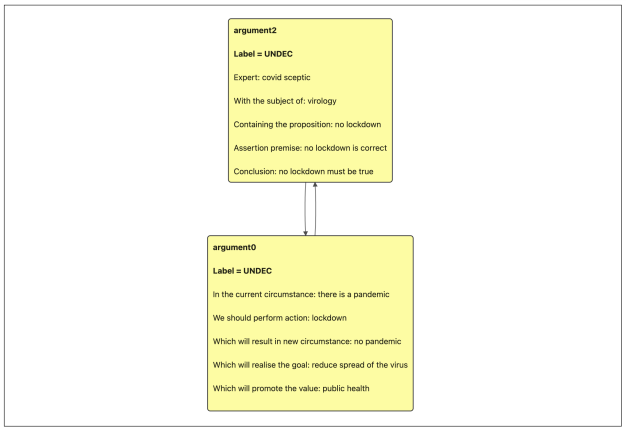
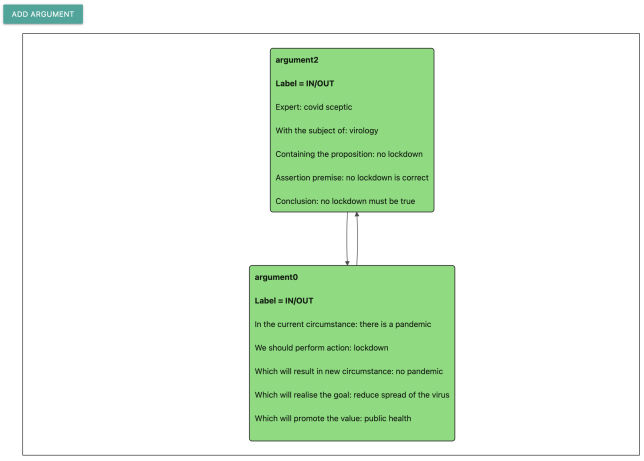
9.4 Completed requirements and figures

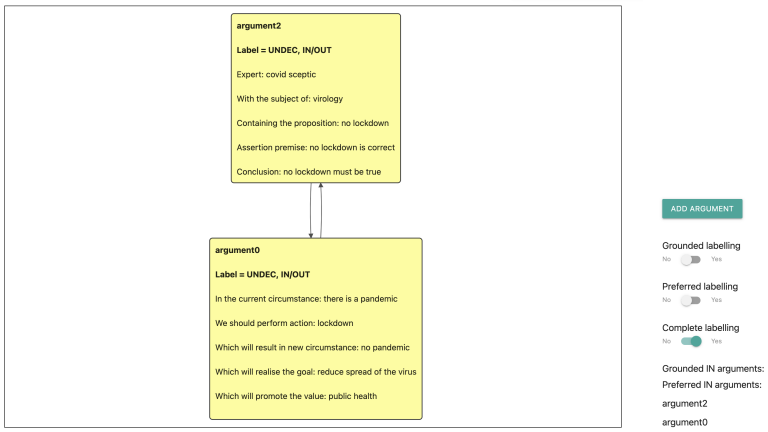
Requirements	1 - *minimum* - Updatable back-end database
Priority	- High
Completed	- Yes, with Firestore database pictured below for arguments and links.
Screenshot	

Requirements	2 - *minimum* - Ensure argument input conforms to schemes
Priority	- High
Completed	- Yes, pictured modal adapts on the basis of argument and scheme chosen by pickers.
Screenshot	

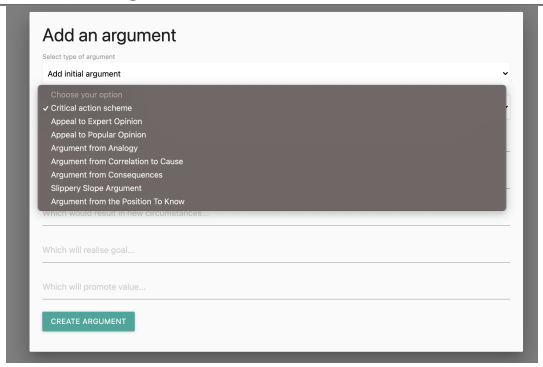
Requirements	3 and 4 - *minimum* - Enable initial arguments and counter-arguments
Priority	- High
Completed	- Yes, users are presented with counter-argument critical questions and graphically counter-arguments links can be expressed with and without conflicting terms.
Screenshot	   ADD ARGUMENT  ADD ARGUMENT

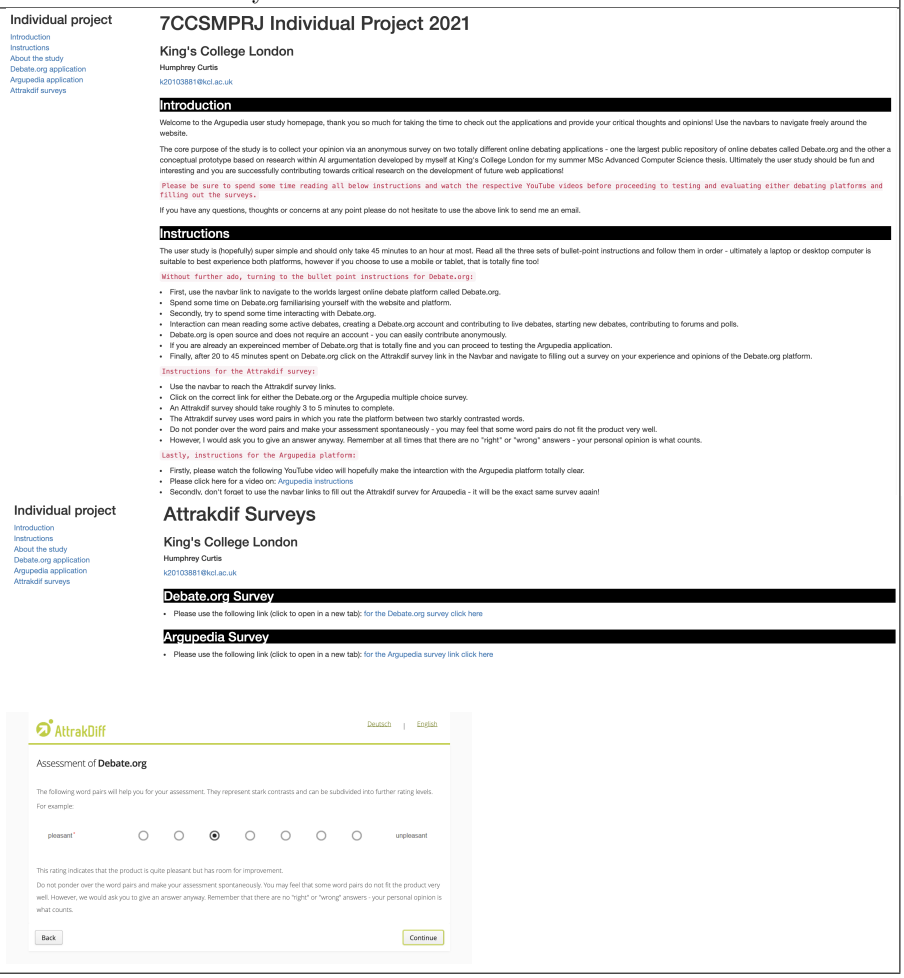
Requirements	5 - *minimum*- Graphical visualisation of argumentation graphs
Priority	- High
Completed	- Yes, argument graphs pictured below.
Screenshot	

Requirements	6, 7 and 8 - *minimum* - Grounded and preferred labelling algorithms
Priority	- High
Completed	- Yes, switches to denote labelling sought for user and effective use of colouring IN (green), UNDEC (yellow) and OUT (red).
Screenshot	  

Requirements	9 - *extended* - Complete labelling algorithm
Priority	- High
Completed	- Yes, switches and labelling provided.
Screenshot	

Requirements	10 - *extended* - Learnable GUI
Priority	- Medium
Completed	- Yes, YouTube video provided to UX participants.
Screenshot	

Requirements	11 - *extended* - Further argument schemes
Priority	- Medium to High
Completed	- Yes, 8 argument schemes offered in total.
Screenshot	

Requirements	12, 13 and 14 - *extended* - Equip Argupedia for UX testing
Priority	- High
Completed	- Yes, full documentation and helpful links provided to participants coupled with Attrakdiff surveys.
Screenshot	

Requirements	15 - *extended* - Employ a voting algorithm
Priority	- Medium to low
Completed	- No

Requirements	16 - *extended* - Have a login
Priority	- Medium to low
Completed	- No

Requirements	17 - *extended* - Extend interface for easy adding of more schemes
Priority	- Medium
Completed	- No

9.4.1 Reflection on development

Development was broadly successful. All minimum requirements were successfully completed. On top of this, over half of the extended requirements were also successfully developed. More requirements could have been achieved and I could have built a more feature rich Argupedia application - however, user testing was ultimately prioritised thereby restricting the time period for development. Ultimately, the user testing was very successful and the prototype Argupedia performed very well versus a market leading competitor. In addition, in all stages of UX testing participants found Argupedia's interface very usable.

Chapter 10

Surveys, test protocol and software testing

This chapter firstly discusses Attrakdif surveys - the survey of choice for UX testing Argupedia. Before secondly, discussing the user-experience A/B test protocol for the Argupedia web application. Finally and thirdly, this chapter briefly considers the white box versus black box test strategies employed during development.

10.1 AttrakDiff surveys

To achieve an accurate reflection on user experience with Argupedia and the control argumentation platform - Debate.org, I chose to use an existing evaluation questionnaire software called AttrakDiff, pictured in figure 10.1. This software allows users to accurately select how they felt using a sliding scale between two word pairs. This approach critically removed bias as founder of the AttrakDiff method, Hassenzahl notes, "a poet may find beautiful words to describe her experience, this does not make it superior to what mundane people experience" [30].

The data could be accurately analysed to compare user experiences between Argupedia and Debate.org - perfect for resolving hypothesis 1 - as to whether the Argupedia application was more user friendly and engaging than Debate.org. Indeed, Attrakdiff surveys adapt to human nature by collecting both pragmatic and hedonic qualities. Pragmatic quality refers to the products ability to complete tasks, whereas hedonic refers to the users relation to the product and its contribution to wards a positive or negative experience.

Hassenzahl is deeply critical of most UX models of evaluation which are focused on only pragmatic qualities, examining how well a product performs a particular task rather than how the user feels as the task is being carried out [30]. The Attrakdiff questionnaire software will collect for both variables - the hedonic aspect is an important component of the design of Argupedia because it is essential to leave a positive lasting impression and to have users to want to keep using the platform.

Lastly, for hypothesis 2 all participants self-reported honestly as to which platform they felt taught them most about the philosophies of argumentation, logic and debate. In essence, which platforms provided the best educational experience.

10.2 A/B user experience testing

In light of the hypotheses noted in chapter 6, Argupedia was user-tested versus the current most popular online argumentation platform - Debate.org. A/B testing was chosen because it offered a simple controlled experiment with fast and effective results. Indeed, large social media sites have frequently favoured using A/B UX testing to make their online products more successful [67].

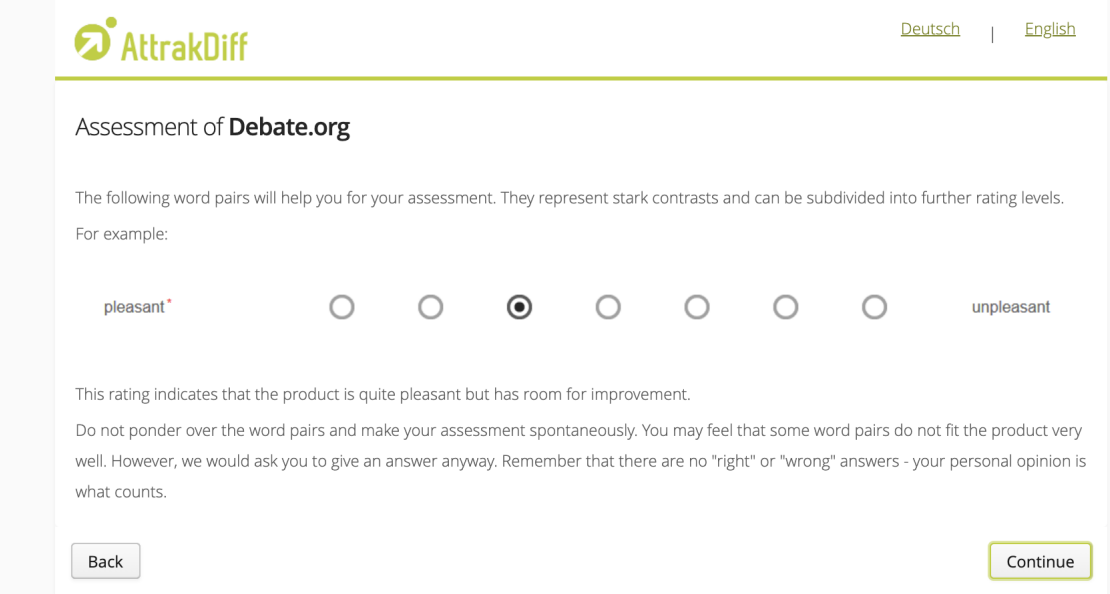


Figure 10.1: Attrakdiff surveys structure with voting between two diametrically opposed word pairs.

Further support for A/B testing comes from Nielsen whose impactful research found 5 users to be the optimum number for finding most usability problems with any software [10]. Given this, from the outset I aimed to have a small pilot study of 3 to 5 users A/B test a pilot version of the Argupedia platform - rather than a larger number of test subjects.

Other advantages of A/B testing include - directly comparing Argupedia with the leading market rival as control would provide clear evidence for Argupedia's weaknesses and therefore offer the best feedback for improvement [67]. Furthermore, as Argupedia is such a bold innovation - it was also good to test it against a market leading counterpart. Notably, a minor potential weakness of A/B UX testing Argupedia was perhaps that it took up relatively significant resources and time [67].

10.3 Testing protocol

Due to the constraints of the pandemic testing took place at long distance and not in person per King's College London's minimal ethical risk testing protocols. All instructions were enclosed and documented within the Argupedia web-application so that test participants could work through the UX test at their own pace for 45 minutes to just over an hour. Testing for each participant was split into eight stages:

- Initially, participants were offered a meet and greet over Zoom - a brief two minute session in which I provided guidance and answered any immediate questions.
- Then participants used a link to navigate to the hosted Argupedia web application.
- Participants would read instructions for Debate.org taking roughly 2 minutes.
- Participants could use Argupedia to externally navigate to Debate.org and spend 20 to 30 minutes familiarising themselves with the platform or directly contributing to live debates.
- Participants would then return to Argupedia and use the navbar to complete an Attrakdif multiple choice survey which would take 3 to 5 minutes to complete.

- Lastly, participants would return to the Attrakdif web page and read instructions for Argupedia. Also here they could watch the 6 minute informative YouTube video I made on how to use Argupedia. In total this would take 10 minutes.
- Following this participants could navigate to Argupedia and spend 20 to 30 minutes debating directly on the platform.
- Participants would then lastly return to the Argupedia web page and fill out an identical Attrakdif survey for 3 to 5 minutes concerning Argupedia.
- Participants had completed testing and I was on hand to answer any further questions and some participants informally provided their thoughts.

10.4 Software testing

Software testing for Argupedia can be divided into white and black box testing. White box testing was straightforward with unit testing of functions performed during development and accessible in the JavaScript accessory files for execution.

In contrast, black box testing centred on identifying flow-graphs of user journeys and potential goals of Argupedia users on the platform using the state diagram denoted in 6.8. Following this, I then simulated those actions to make sure that the Argupedia platform performed as expected enabling the user to reach their goals. From these simulations using the state diagram, I made a full black box and flow graph testing report, which I then performed on the Argupedia prototype and for the results of this testing please refer to the appendix report available here: A.8.

On top of this, Argupedia performs some light testing of user input to make sure it is valid and provides a necessary warning to users on incorrect argument inputs. Ultimately, further software and stress testing of the Argupedia platform is needed - however, I simply did not have the time to do this further development and it did not rank too high on my list of overall requirements versus making Argupedia more feature rich.

Chapter 11

Evaluation and graph examples

This chapter describes, evaluates and critiques the final prototype of Argupedia. Following this, I discuss and evaluate some argumentation graph examples generated with the application.

11.1 The final prototype

The final Argupedia web application is a dynamic web environment that provides 6 primary controls utilising buttons and switches. The user can use these buttons to learn more about Argupedia, add arguments and evaluate those arguments with labelling algorithms based on research within AI argumentation - it is pictured here in figure 1.1.

Users can add arguments and counter-arguments utilising eight argument schemes. Upon creating a counter-argument, users are provided with dynamically generated critical questions from the opposing arguments variables and can specify whether the claims are conflicting as pictured in figure 11.3. This ensures that the arguments conform to principled argument schemes and theory. For all arguments entered into the Argupedia repository a dynamic zoom-able and movable argumentation graph is updated. The argumentation graph clearly displays the inter-relation of arguments and counter-arguments - showing how arguments evolve over time. Equally, the users can analyse the graph nodes containing data about each argument.

In addition, Argupedia users can toggle three switches for three different labelling algorithms: grounded, preferred and complete to re-draw the graph with appropriate colours and labels. Users can utilise these argument labellings to evaluate winning arguments during debates and the arguments that are most preferred. To improve accessibility winning arguments are additionally listed within the environment. Now I will evaluate the final prototype discussing weaknesses and strengths.

11.2 Weaknesses

Due to the fact that Argupedia is a prototype coded by just one researcher in limited time there are obviously shortcomings, which would need to be solved in future prototypes of Argupedia. Three key weaknesses of Argupedia include, the prototype is not fully stress tested, future prototypes need to provide more education on debate and argumentation and lastly, more features need to be adopted.

Firstly, with regards to the current Argupedia prototype - little testing or stress testing has taken place. Future versions of Argupedia will need to be capable of hosting many hundreds of users. Therefore, the web application will need to have sufficient cyber security and resources to host plenty of users successfully. In addition, I have not tested all possible forms of data entry from users - therefore future versions of Argupedia will have to do more testing on users data entry to ensure that the web application is robust and not glitchy or breakable.

Secondly, the current Argupedia prototype offers a YouTube video I made to educate users and helpful links to further academic resources explaining the AI argumentation foundations

Figure 11.1: YouTube video created to provide instructions for users of Argupedia.

Figure 11.2: Argupedia navigation bar with buttons providing links to further information.

of the platform as depicted by figures 11.1 and 11.2. Nonetheless, future versions of Argupedia must provide more detailed and customised tutorial, information on the purposes of the platform and insightful academic material. At present, a start has been made on Argupedia's documentation, tutorials and academic foundations - yet it would be good to continue in this vein and provide a really clear understanding of the purposes and foundations of the platform.

Thirdly, the current Argupedia prototype as per the specification offers plenty of features yet its exciting to think of future directions for Argupedia with more features. For Argupedia, it would be useful to have a formal user login with a user profile page showcasing the argument graphs that the user has been contributing towards. Additionally, ensuring Argupedia provides multiple threads of graphs and perhaps even a homepage offering useful analytics on the current most popular graphs, labelling algorithms, schemes and debates on the platform. Now I will turn to the strengths of the Argupedia platform.

11.3 Strengths

The initial weaknesses of Argupedia can be largely attributed to the device being a prototype developed single-handedly over a short period of time - resulting in a not completely polished product. Despite this, there are plenty of strengths concerning the presented Argupedia prototype, however for the purpose of evaluation I would like to emphasize three core strengths. These strengths would be: an interactive user interface, intuitive controls and a seamless argument repository.

Firstly, the Argupedia interface is very interactive yet minimalist with keystrokes and commands providing high levels of user control and manipulation as denoted by figures 11.3, 11.4, 11.5 and 11.6. In particular, the display of the argumentation graph is highly interactive with users able to zoom (scroll on a mouse or push commands on track pad) and manipulate the graph in state space - enabling maximum user engagement with the argument graphs and providing users with the opportunity to deeply study the graphs.

Secondly, the current Argupedia prototype offers intuitive controls - users are presented with

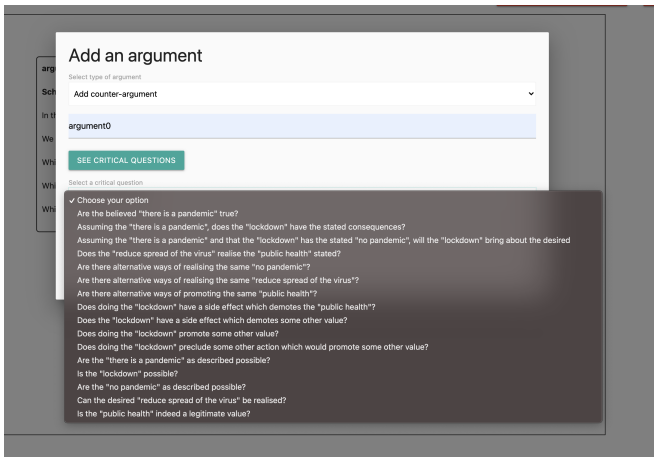


Figure 11.3: Critical questions dynamically presented after having selected an argument target.

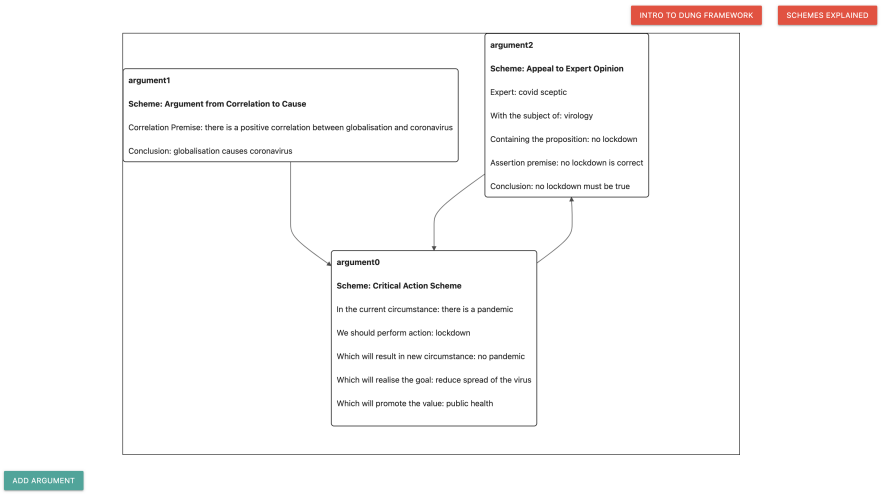


Figure 11.4: Graphs generated within the Argupedia application.

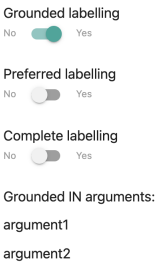


Figure 11.5: The grounded labelling switch applied.

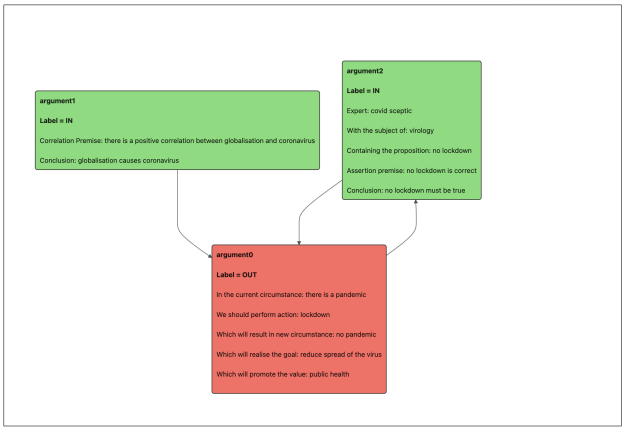


Figure 11.6: Arguments denoted in green (IN) and red (OUT) under the grounded labelling.

buttons and switches in the same vein as other applications familiar to their mental models as denoted by figure 11.5. When users click buttons to create arguments - the application does not provide all form information and perpetuate confusion. Instead, Argupedia slowly unfolds the form as choices are successfully made by the user - ensuring that all arguments submitted to the platform are totally valid. For example, the critical questions appear dynamically and immediately upon command as denoted by figure 11.3.

Thirdly and lastly, the current Argupedia prototype performs very well and robustly with the back-end database and repository. Throughout all testing no problems were encountered with seamless storage and latency issues. In addition, when tested with weak WiFi - Argupedia still performed well, with local storage in the browser. Furthermore, the database is fully extendable for future development.

11.4 Showcase argumentation graphs

The following figures from 11.7 onwards are from participants debating on the platform during UX testing as well as examples are captured in screenshots below. Further debates were had by participants but the file size of this thesis is capped.

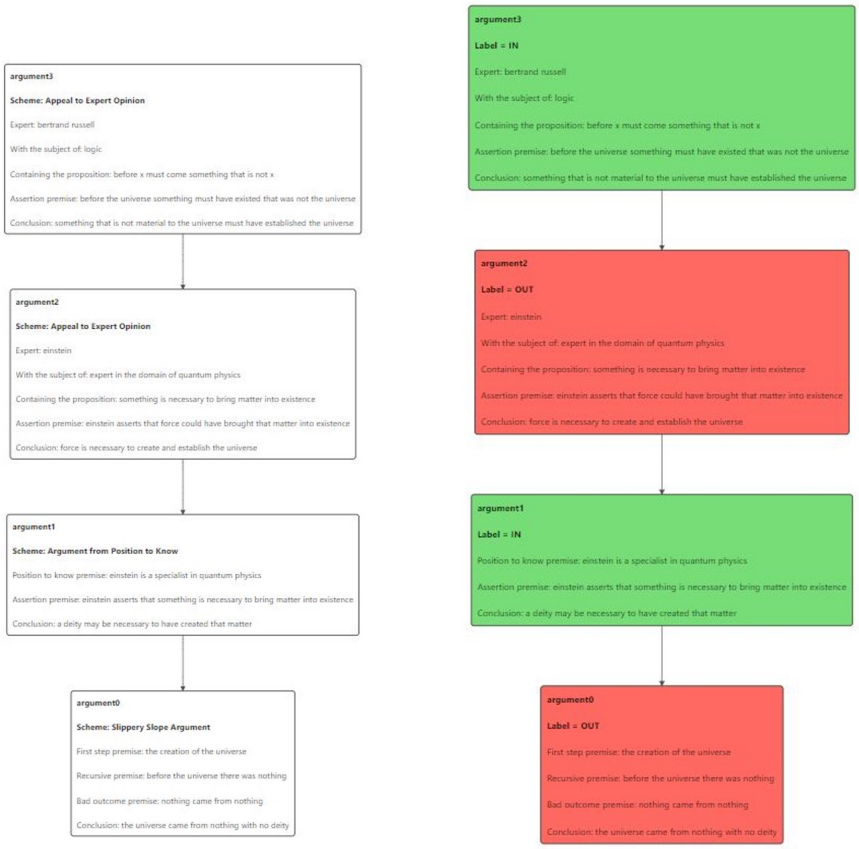


Figure 11.7: An argument graph created by users during formal UX testing, here the graph forms a long chain of arguments considering whether God created the universe under the grounded labelling. Chaining of arguments in a simple graph was common when users were initially inexperienced with the Argupedia platform.

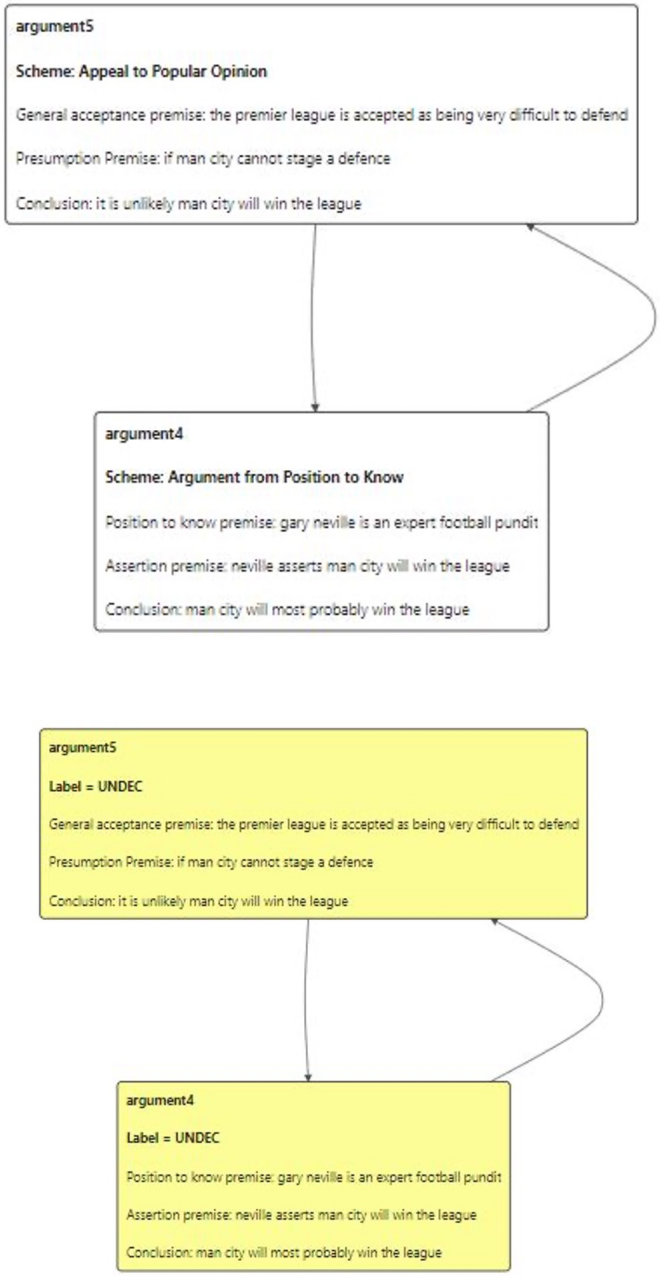


Figure 11.8: An argument graph created by users during formal UX testing with conflicting claims considering whether Man City will win the Premier League in 2021/2. Both arguments are UNDEC under the grounded labelling [8].

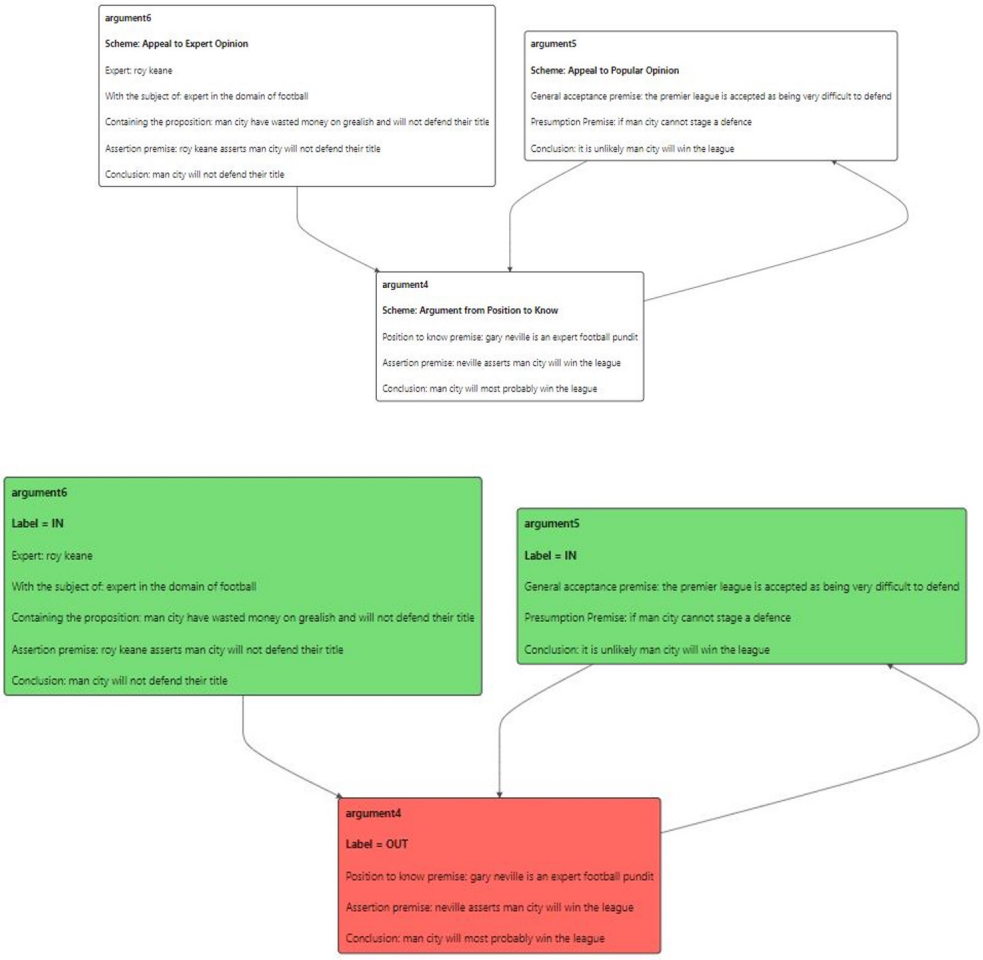


Figure 11.9: A user added an argument citing Roy Keane to shift the status of the debate in favour of Man City not defending the Premier League - also depicted under the grounded labelling.

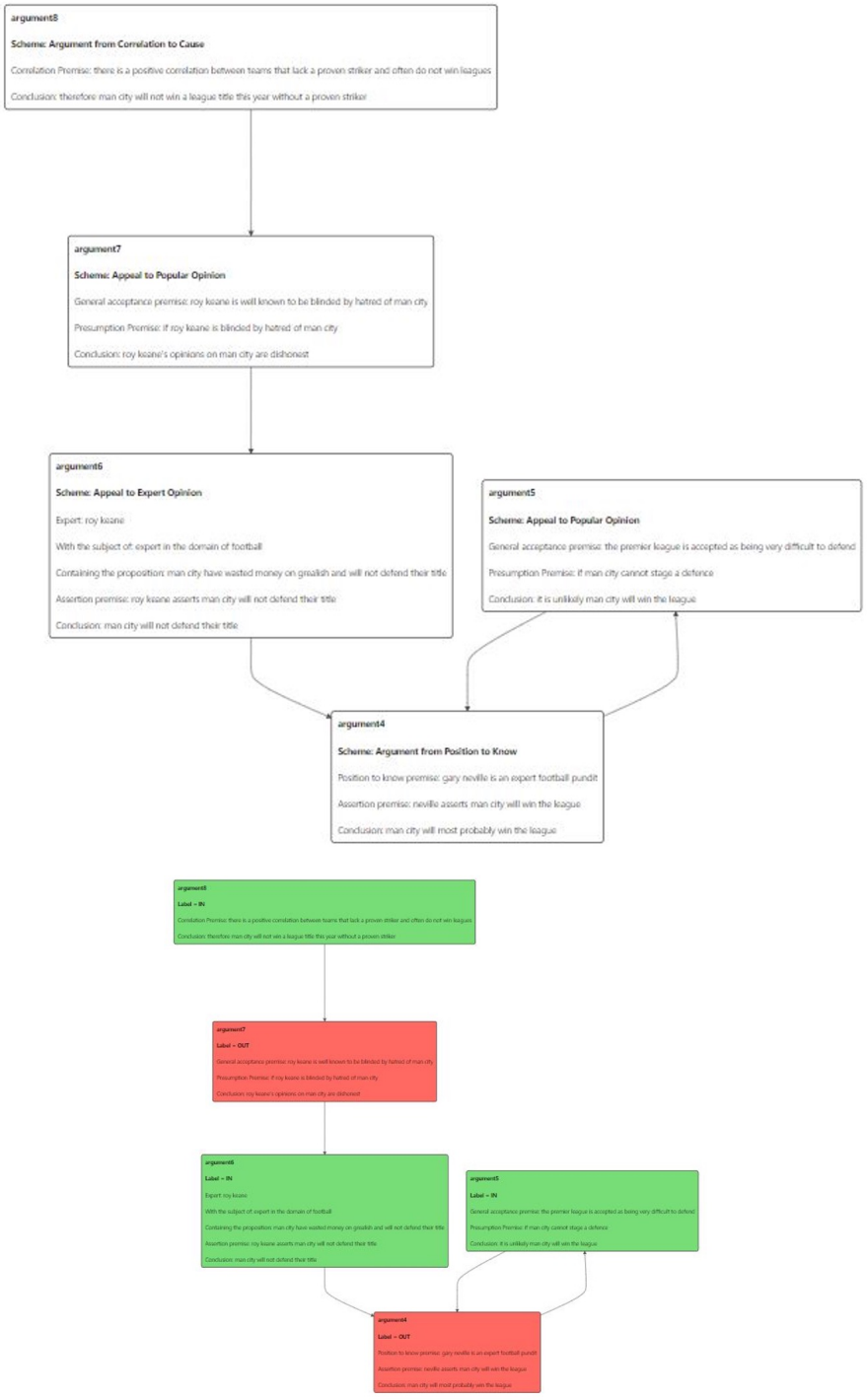


Figure 11.10: Ultimately, the argument that Man City lack a striker was decisive in shifting the debate that Man City will not win the league.

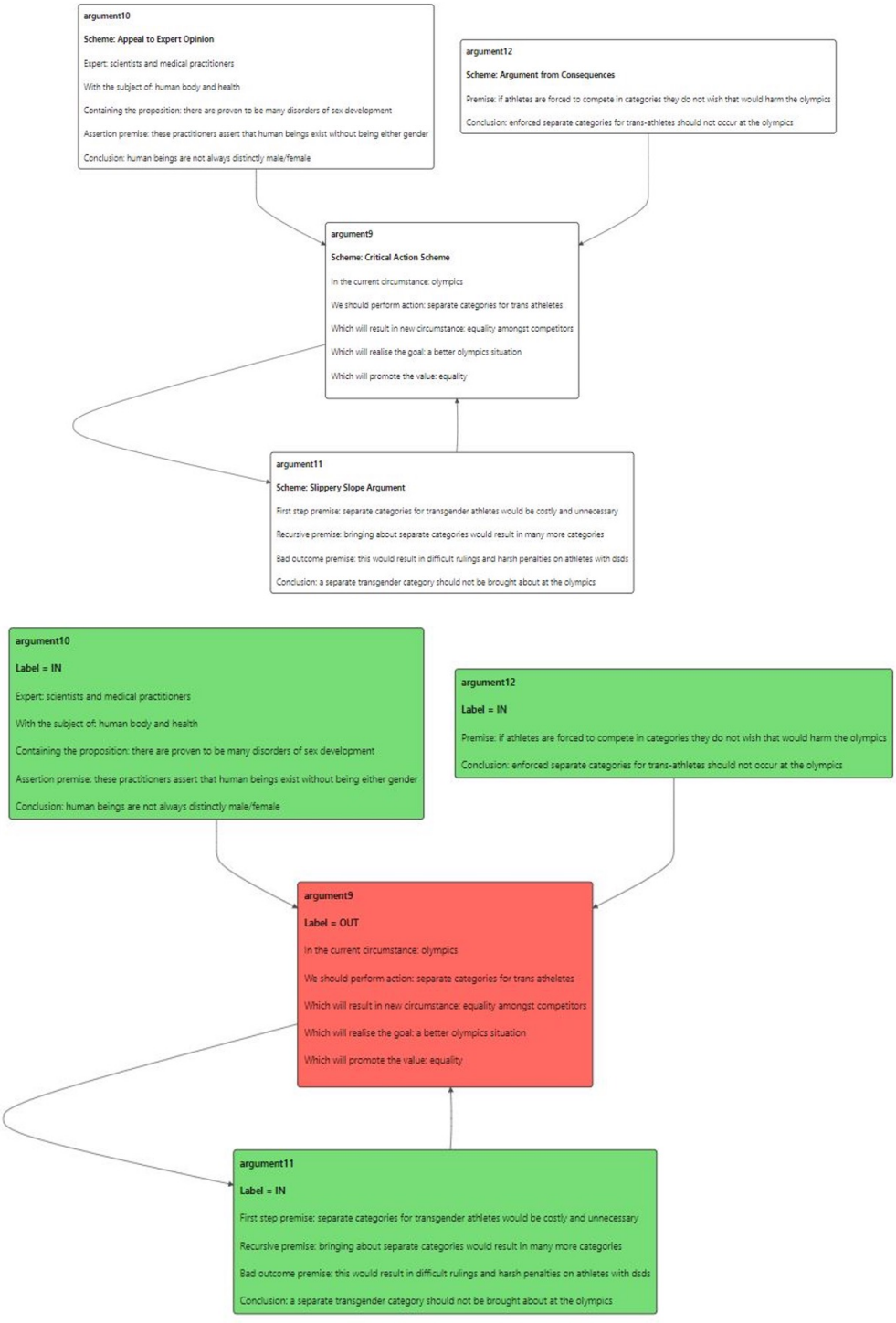


Figure 11.11: During UX testing a one-sided debate graph was observed. Here, the debate graph under the grounded labelling strongly supports the claim that there should *not* be a separate category for transgender athletes. This was in light of public discourse concerning female athlete Laura Hubbard at the 2021 Olympics [68].

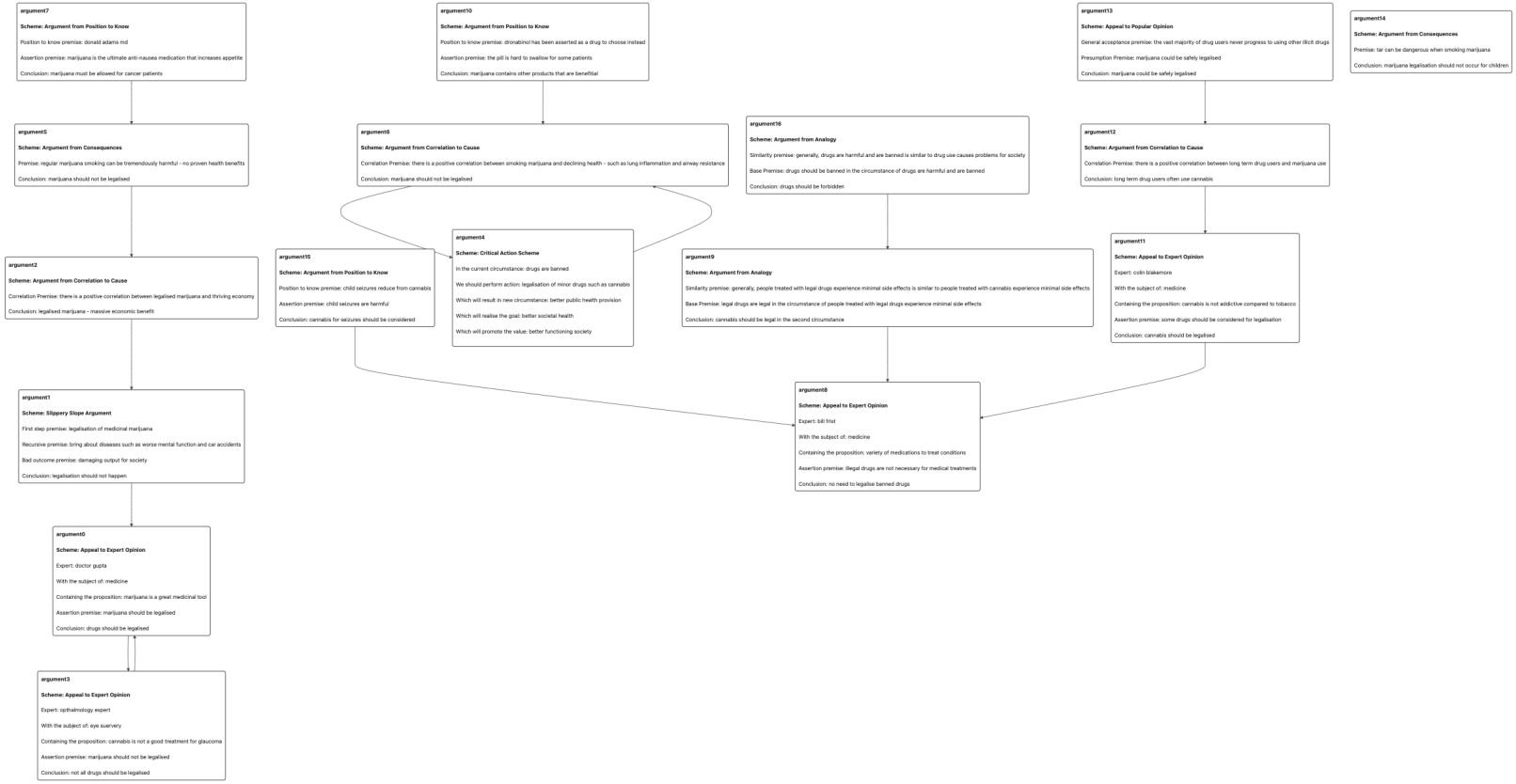


Figure 11.12: Large 16 argument debate graph created and added too during UX testing. The debate considered legalisation of drugs and marijuana.

Figure 11.13: Large debate of 16 arguments under grounded labelling.

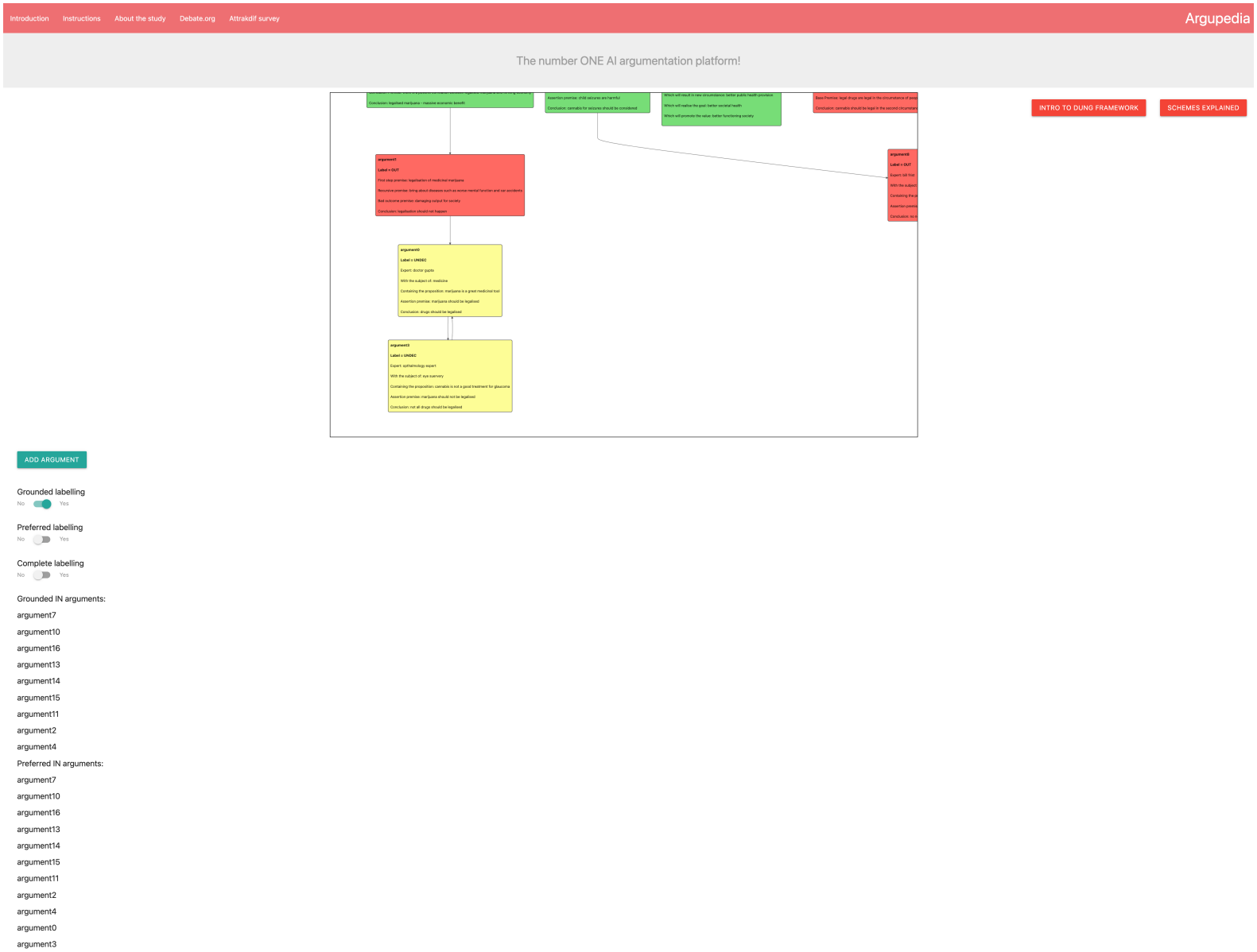


Figure 11.14: Using the Argupedia UI to successfully zoom and closely study the arguments on the large Debate graph.

Chapter 12

Results

This chapter will discuss the results of UX testing of Argupedia with four participants in the initial pilot trial. Initially, questionnaire data will be discussed - providing an insight into the performance of Argupedia versus the control debate platform, namely Debate.org. Following this, I will discuss some of the verbal qualitative feedback generously provided by one of the experienced participants on Argupedia and whether my initial two hypotheses were validated. Lastly, I will provide an overview and evaluation of the results and limitations of the study.

12.1 Attrakdiff questionnaire results

For reminder, Pragmatic qualities (PQ) refers to the usability of the interface. Hedonic qualities (HQ) are divided into stimulation (HQ-S), which refers to how novel and interesting the interface experience, and identity (HQ-I), which refers to the extent which a user can identify with the interface. Finally Attractiveness (ATT) describes a global value of attractiveness based on the interface's quality. Across all participants the scores within the key categories are pictured in figures 12.1, 12.2 and 12.3.

Pragmatic qualities (PQ) - Debate.org scored a respectable amount for PQ values as to be expected from the worlds largest online debate platform. In contrast, Argupedia perhaps suffered as it is a prototype and perhaps was slightly overly technical - in the next section on additional qualitative feedback useful advice is given to make future versions of Argupedia more pragmatic.

Hedonic qualities interaction (HQI) - Debate.org scored low for this category - the adversarial nature of the debate platform may be damaging for users ability to identify with the platform. Whilst, Argupedia scored respectably in the category perhaps due to the more constructive nature of debating and minimalist UI.

Hedonic qualities stimulation (HQS) - Debate.org scored very low in this category - it is not a particularly novel platform. In contrast, Argupedia comes in to its own scoring exceptionally high - many praised its bold and novel purposes for debating and learning about argumentation.

Attractiveness (ATT) - Debate.org scored respectably in this category with many finding it pleasant to use but a slightly outdated and ugly experience. Similarly, Argupedia scored respectably but slightly higher - participants preferred its minimalist UI albeit perhaps lacking colour.

12.2 Additional qualitative participant feedback

All four participants self-reported that they learnt more about debate and argumentation from using Argupedia versus the control, Debate.org. In particular, they emphasised that learning about the schemes, argument labelling algorithms and critical questions improved their ability to think critically and argue in a cohesive and positive manner. Furthermore, adding to this

positive data one participant - an advanced university and international level debater offered some reflective commentary during an informal interview after his UX study.

Starting with criticisms, the participant noted that he wanted to be firstly met with a prototypical "long argument graph" to immediately demonstrate the potential of the Argupedia platform. Additionally, he felt that the argument scheme text could be "more clear visually" within the Argupedia prototype. Despite the further information buttons, he even suggested that for each scheme an "additional pop-up should present an exemplar argument to the user for each argument scheme" - to really present users with an example of how to use each scheme successfully within Argupedia. At times, he felt slightly "unsure how to use schemes" despite the further information provided by the UI. In contrast, the participant was impressed with Debate.org's "active user base" versus Argupedia and felt that Argupedia should eventually have a similar variety of clearly defined topics and discussion (e.g. religion, politics, etc.), which are equally well-argued by the user-base. At times, he felt that the Argupedia's prototype UI was "slightly clunky" and he believed that it would be better served with slightly "more colour" and imagery to prevent user "frustration". Lastly, he said he would have liked to have a voting mechanism on the prototype Argupedia platform.

Turning to positives of Argupedia, the participant felt that the restriction to only using argument schemes and critical questions to debate was an "amazing concept". Despite his previous debating experience, he believed that it really taught him "a lot" about argumentation and debate. Versus Debate.org, the participant thought that Argupedia's UI and "presentation of debates was incredible" - he loved the "zoom" feature and "highly intuitive way" of viewing debate graphs coupled with the interrelation of arguments clearly depicted by the graph links - he felt that this advanced feature of Argupedia served as a "significant advantage over Debate.org". Looking to the future - the participant stressed that these features of Argupedia would be very significant because it meant that he was having arguments that culminated in "rational debate" and "constructive conclusions" with "both sides of the argument validated by Argupedia's labelling algorithms". In contrast, on Debate.org he self-reported that arguments were too often "highly adversarial" meaning that two users were purely arguing in "bad faith" as a strategy "for the sake of winning debates versus an adversary" - whereas on Argupedia he felt that this was not the case and it was a much more "constructive debate platform".

12.3 Evaluation of findings and limitations

The study was conducted over a week period remotely with 4 participants - 2 male and 2 female. With respect to hypothesis 1, the outcome was mixed. Participants did not find Argupedia more user-friendly yet counter-intuitively they found Argupedia more engaging than the control, Debate.org. Across the 4 participants Argupedia scored lower than Debate.org for pragmatic qualities. Despite this, Argupedia scored significantly higher than Debate.org for hedonic qualities - suggesting that Argupedia is a more engaging yet slightly more complicated and confusing than Debate.org.

In contrast, with respect to hypothesis 2, all four participants self-reported that they accrued more knowledge about the philosophies of argumentation, logic and debate from having used Argupedia versus Debate.org. Indeed, Argupedia is certainly a much more academic debating tool versus the control which is more adversarial in the vein of social media. So hypothesis 2 was verified by the user-study.

Nonetheless, there are some clear limitations of the study. Firstly, participants were not sampled randomly - two of the participants were expert level debaters. Secondly, the UX testing did not take place in a controlled setting such as a laboratory. Thirdly, all participants had received higher education and thus did not offer a fair reflection of a normalized population.

In response to these limitations, I did not have a project budget to fund a suitably expensive UX study to mitigate these problems. Furthermore, the pandemic made any in-person user testing challenging and pushed the testing environment outside of controlled settings. Therefore, it was good to design an app that participants could effectively use and test remotely. Ultimately, the UX testing was very successful with all four participants engaging with the platforms and providing data successfully for both hypotheses.

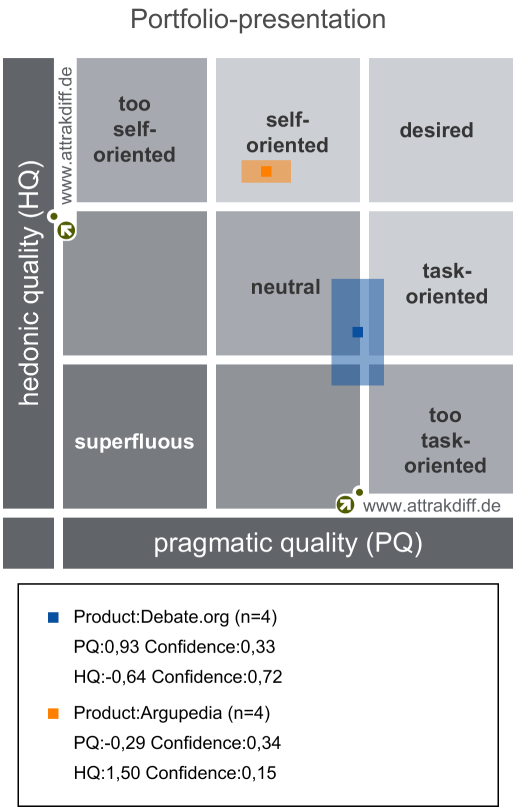


Figure 12.1: Overall results from surveys - Argupedia performed much better than Debate.org in hedonic quality yet was comparably weaker in pragmatic qualities. Nonetheless, Argupedia is not far from a desired position.

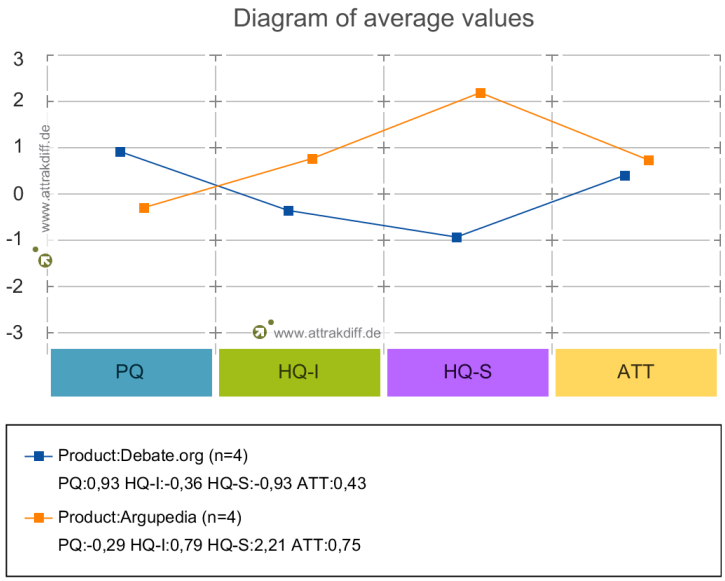


Figure 12.2: Argupedia performed well in testing achieving higher rankings than Debate.org across 3 out of 4 categories.

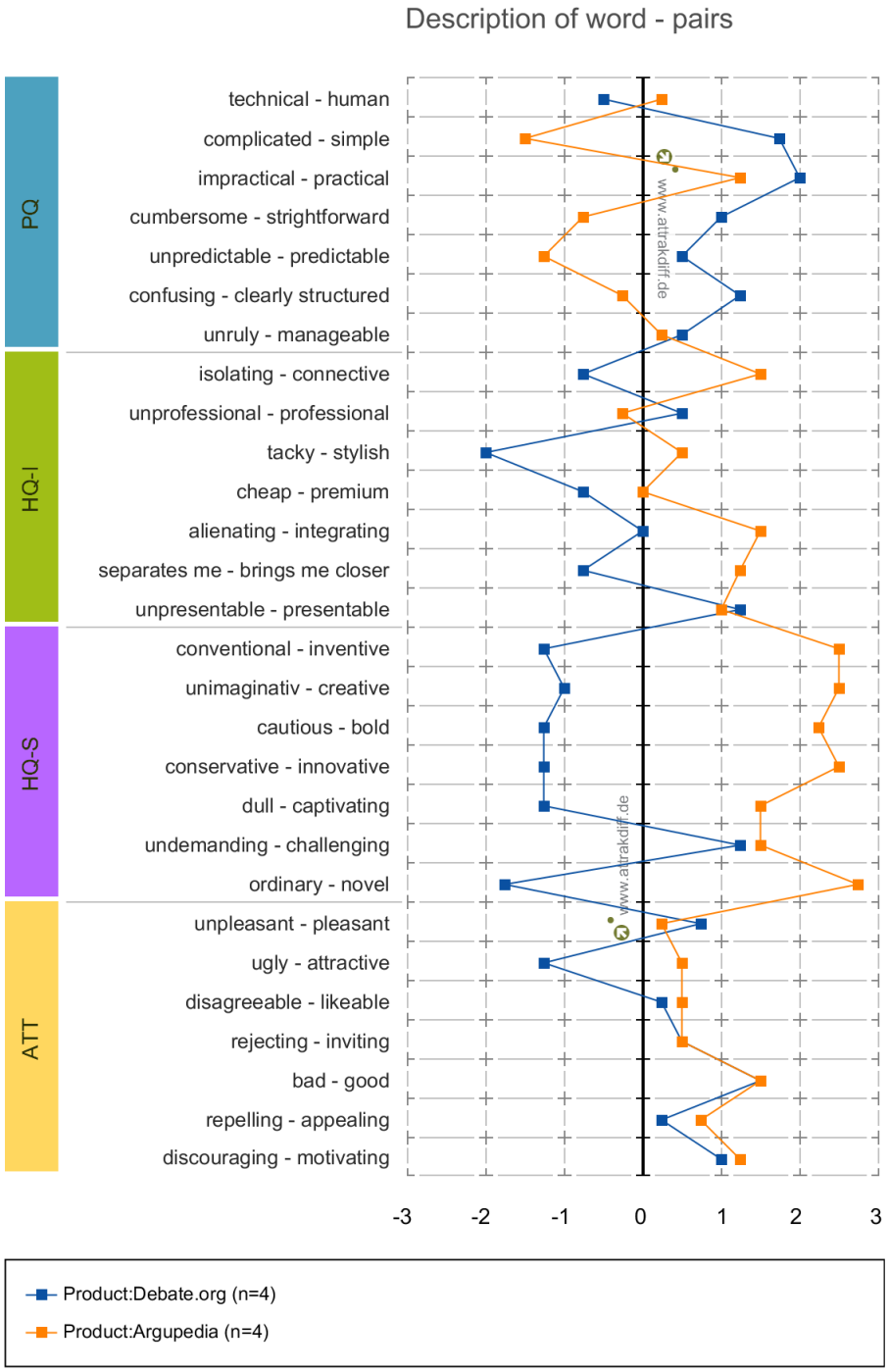


Figure 12.3: From word pair results Argupedia is found to be more complicated, impractical and confusing versus Debate.org. However, Argupedia is also significantly more inventive, bold, creative and innovative.

Chapter 13

Conclusion and future work

13.1 Findings

With respect to the project motivations and objectives outlined at the start of the thesis (chapter 1), the Argupedia project has critically taxonomised insights from Dung's abstract argumentation (chapter 2), schemes and critical questions (chapter 3) and GUI design plus HCI principles (chapter 4) as well as provided analysis on rival argumentation platforms (chapter 5). From the literature review and rival analysis this study designed a USP for a new online debating platform called Argupedia.

Argupedia is a novel interactive repository designed to capture how public debate evolves. Argupedia utilises Dung argumentation graphs to provide users with the ability to structure relational arguments via argument schemes. Critically Argupedia at all times provides users with a contextual understanding of the most preferred arguments according to labelling algorithms. This mitigates weaknesses of rival platforms - enabling users to structure arguments rationally, counter-argue effectively and get a clear understanding of the status of the debate. Furthermore, this study has outlined Argupedia's design and construction period, including evaluation of the software and eventual prototype (chapters 6 to 11).

Initial findings from a pilot UX study (chapter 12) suggests that Argupedia performs very well versus a market leading competitor - namely, Debate.org. With respect to the two hypotheses, Argupedia excelled at providing users with an engaging and novel interaction experience albeit at the expense of some minor usability quirks. Furthermore, all 4 participants self-reported improvements in their understanding of the philosophies of argumentation, logic and debate versus the control. Additionally, I have provided key qualitative insights from an experienced participant (an internationally recognised debater) to improve future Argupedia prototypes. Lastly, for the project to be fully complete, I will now suggest pathways for future research.

13.2 Suggestions for future work

After the Coronavirus pandemic has subsided, further stress and user testing is recommended with Argupedia - particularly in controlled laboratory settings to collect further data from users feedback and experience with the application. Ultimately, the Argupedia front-end can perform slightly buggy due to a lack of fully fledged testing. Following this, there are several opportunities to develop Argupedia further:

Incomplete extended requirements - Deliberately some of the extended requirements are incomplete due to lofty initial goals and to provide a basis for future work. The incomplete extended requirements are providing a voting algorithm, supporting arguments, having a user login and a back-end web page to enable easy upload of future schemes. All these features have advantages - voting enables a more democratic platform, supporting arguments provides another feature to strengthen pre-existing arguments position, a user login enables personalisa-

tion and users to have a backlog of arguments and a back-end web-page provides accessibility for laypeople to upload argument schemes. Based on the current code-base - all these features would be able to be applied quickly.

More documentation and academic information - Requested during UX testing from participants and to improve the Argupedia experience. More tailored documentation and academic information concerning argument graphs, schemes and critical questions would be very beneficial. The participant informally interviewed, suggested Argupedia should have pop-ups which provide exemplars of each argument scheme. Perhaps, Argupedia could even have a separate page with customised videos breaking down the academic underpinnings and providing more information for users interested in furthering their knowledge of argumentation and debate.

Development of a homepage - Argupedia could be expanded upon with an innovative homepage in the model of other participation websites like Reddit, Stackoverflow and others. The Argupedia homepage can serve as a directory and present to users useful analytics such as active debate graphs, popular debate graphs, most up-voted arguments and different threads of discussion or debate (e.g. arts, economics). As well as suggested graphs and content for returning users with an account for Argupedia already.

Machine and deep learning - Argupedia offers tremendous potential for machine and deep learning techniques in two key ways. Firstly, Argupedia's repository of data can serve as a resource to inculturate and educate artificial intelligence [48]. Indeed, Argupedia could even be developed to actively debate and educated users using the repository of data contained within the application. Secondly, Argupedia's front-end can be improved with AI and data query techniques. For instance making the front-end more accessible and customised based on pre-existing user data and preferences.

13.3 Concluding remarks

At the time of writing, following a two year global pandemic and divisive US Presidential election, polarisation and echo-chambers online regularly through social media continue to grow at alarming rates [35, 52, 58]. However, innovations and platforms which educate their usership and inspire rational dialogues between counter-parties still have yet to become mainstream amongst the public online.

Yet, as demonstrated by Argupedia - users can be: guided to debate rationally, shown how to counter-argue without resorting to fallacious arguments and be provided with a less adversarial depiction of the current arguments which are most preferred. Resulting in a more constructive online environment for all users - no matter their differing opinions.

Indeed, future researchers should remember this going forward, which will hopefully result in the emergence of better structured online platforms becoming publicly available sooner. Increased perforation of constructive platforms such as Argupedia have the capacity to significantly improve the quality of public online discourses, debates and discussions.

References

[1] Solomiia Albota. Resolving conflict situations in reddit community driven discussion platform. In *COLINS*, pages 215–226, 2020.

[2] Aristotle. *On sophistical refutations*. Kessinger Publishing, 2004.

[3] Katie Atkinson and Trevor Bench-Capon. Action-based alternating transition systems for arguments about action. In *AAAI*, volume 7, pages 24–29, 2007.

[4] Katie Atkinson and Trevor Bench-Capon. Addressing moral problems through practical reasoning. *Journal of Applied Logic*, 6(2):135–151, 2008.

[5] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. *The knowledge engineering review*, 26(4):365–410, 2011.

[6] Jordan Beck, Bikalpa Neupane, and John M Carroll. Managing conflict in online debate communities: Foregrounding moderators' beliefs and values on kialo. *SocArXiv*, 2018.

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

[8] Chris Bevan. Premier league predictions 2021-22: Bbc sport pundits pick their top four.

[9] Nigel Bevan. Usability is quality of use. In *Advances in Human Factors/Ergonomics*, volume 20, pages 349–354. Elsevier, 1995.

[10] Nigel Bevan, Carol Barnum, Gilbert Cockton, Jakob Nielsen, Jared Spool, and Dennis Wixon. The" magic number 5" is it enough for web testing? In *CHI'03 extended abstracts on Human factors in computing systems*, pages 698–699, 2003.

[11] John A Burgess. The great slippery-slope argument. *Journal of medical ethics*, 19(3):169–174, 1993.

[12] Martin Caminada. A gentle introduction to argumentation semantics. *Lecture material, Summer*, 2008.

[13] Thomas D Cook, Donald Thomas Campbell, and William Shadish. *Experimental and quasi-experimental designs for generalized causal inference*. Houghton Mifflin Boston, MA, 2002.

[14] Michael E Cope and Kevin C Uliano. Cost-justifying usability engineering: A real world example. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, pages 263–267. SAGE Publications Sage CA: Los Angeles, CA, 1995.

[15] Rikke Friis Dam and Teo Yu Siang. 5 stages in the design thinking process.

[16] John Danaher. Philosophical disquisitions: argumentation schemes, March 2010.

[17] Matthew d'Ancona. *Post-truth: The new war on truth and how to fight back*. Random House, 2017.

[18] Andrew Daniels, Grant Patten, and Alex Whiteman. Debate. org. *University of Toronto*, 2010.

[19] Debate.org. Debate.org wiki.

[20] Martijn H Demollin. The argument from correlation to cause in science communication. *Argument: Biannual Philosophical Journal*, 10(2):315–332, 2020.

[21] Peter J Denning. Design thinking. *Communications of the ACM*, 56(12):29–31, 2013.

[22] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.

[23] Catarina Dutilh Novaes. Argument and Argumentation. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Fall 2021 edition, 2021.

[24] Rod L Evans. Appeal to popular opinion. *International Philosophical Quarterly*, 40(3):387–389, 2000.

[25] John Fox, Nicky Johns, and Ali Rahmanzadeh. Disseminating medical knowledge: the proforma approach. *Artificial intelligence in medicine*, 14(1-2):157–182, 1998.

[26] Wilbert O Galitz. *The essential guide to user interface design: an introduction to GUI design principles and techniques*. John Wiley & Sons, 2007.

[27] Yousef Haik, Sangarappillai Sivaloganathan, and Tamer M Shahin. *Engineering design process*. Cengage Learning, 2015.

[28] Crispin Hales et al. *Analysis of the engineering design process in an industrial context*. PhD thesis, University of Cambridge, 1987.

[29] Richard Mervyn Hare and Richard Mervyn Hare. *Freedom and reason*, volume 92. Oxford Paperbacks, 1963.

[30] Marc Hassenzahl. Hedonic, emotional, and experiential perspectives on product quality. In *Encyclopedia of human computer interaction*, pages 266–272. IGI Global, 2006.

[31] Arthur Claude Hastings. *A Reformulation of the Modes of Reasoning in Argumentation*. Northwestern University, 1962.

[32] Ann Heylighen and Andy Dong. To empathise or not to empathise? empathy and its limits in design. *Design Studies*, 65:107–124, 2019.

[33] Hans Hoeken, Rian Timmers, and Peter Jan Schellens. Arguing about desirable consequences: What constitutes a convincing argument? *Thinking & reasoning*, 18(3):394–416, 2012.

[34] World Leaders in Research-Based User Experience. Putting a/b testing in its place.

[35] Julie Jiang, Xiang Ren, Emilio Ferrara, et al. Social media polarization and echo chambers in the context of covid-19: Case study. *JMIRx Med*, 2(3):e29570, 2021.

[36] André Juthe. Argument by analogy. *Argumentation*, 19(1):1–27, 2005.

[37] John Karat, Clare-Marie Karat, and Jacob Ukelson. Affordances, motivation, and the design of user interfaces. *Communications of the ACM*, 43(8):49–51, 2000.

[38] Kialo. Explore popular debates, discussions and critical thinking.

[39] Manfred Kienpointner. Alltagslogik struktur und funktion von argumentationsmustern. *Frommann-Holzboog*, 1992.

[40] Rochelle King, Elizabeth F Churchill, and Caitlin Tan. *Designing with data: Improving the user experience with A/B testing.* " O'Reilly Media, Inc.", 2017.

[41] Maaike Susanne Kleinsmann. Understanding collaborative design. *Springer*, 2006.

[42] Ron Kohavi and Roger Longbotham. Online controlled experiments and a/b testing. *Encyclopedia of machine learning and data mining*, 7(8):922–929, 2017.

[43] Thomas Kvan. Collaborative design: what is it? *Automation in construction*, 9(4):409–415, 2000.

[44] Kelvin Luu, Chenhao Tan, and Noah A Smith. Measuring online debaters' persuasive skill from text over time. *Transactions of the Association for Computational Linguistics*, 7:537–550, 2019.

[45] Aaron Marcus. Dare we define user-interface design? *interactions*, 9(5):19–24, 2002.

[46] Salisa Maulidiyah et al. Face threatening acts and politeness strategy performed by debaters at debate. org website. *IAIN*, 2016.

[47] Deborah J Mayhew and Randolph G Bias. Cost-justifying usability engineering for cross-cultural user interface design. *Usability and internationalization of information technology*, pages 213–249, 2005.

[48] Sanjay Modgil. Dialogical scaffolding for human and artificial agent reasoning. In *AIC*, pages 58–71, 2017.

[49] Sanjay Modgil and Martin Caminada. Proof theories and algorithms for abstract argumentation frameworks. In *Argumentation in artificial intelligence*, pages 105–129. Springer, 2009.

[50] Virginia Morini, Laura Pollacci, and Giulio Rossetti. Toward a standard approach for echo chamber detection: Reddit case study. *Applied Sciences*, 11(12):5390, 2021.

[51] Leonard Nelson. The socratic method. *Thinking: The Journal of Philosophy for Children*, 2(2):34–38, 1980.

[52] An Nguyen and Hong Tien Vu. Testing popular news discourse on the "echo chamber" effect: Does political polarisation occur among those relying on social media as their primary politics news source? *First Monday*, 24(5), 2019.

[53] Jakob Nielsen. Ten usability heuristics, 2005.

[54] Jakob Nielsen and Rolf Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256, 1990.

[55] Reinhard Oppermann. User-interface design. In *Handbook on information technologies for education and training*, pages 233–248. Springer, 2002.

[56] Andrea Pazienza, Stefano Ferilli, and Floriana Esposito. Constructing and evaluating bipolar weighted argumentation frameworks for online debating systems. In *AI³ @ AI* IA*, pages 111–125, 2017.

[57] Chaim Perelman. The new rhetoric. In *Pragmatics of natural languages*, pages 145–149. Springer, 1971.

[58] Erik Peterson, Sharad Goel, and Shanto Iyengar. Echo chambers and partisan polarization: Evidence from the 2016 presidential campaign. *Unpublished manuscript. https://5harad. com/papers/selecfive-exposure. pdf*, 2018.

[59] John L Pollock. Defeasible reasoning. *Cognitive science*, 11(4):481–518, 1987.

[60] Henry Prakken. Ai & law, logic and argument schemes. *Argumentation*, 19(3):303–320, 2005.

[61] Iyad Rahwan and Guillermo R Simari. *Argumentation in artificial intelligence*, volume 47. Springer, 2009.

[62] Christof Rapp. *Aristotle's rhetoric*. Kessinger Publishing, 2002.

[63] Reddit. reddit.com.

[64] Bruce A Reinig, Robert O Briggs, and Jay F Nunamaker. On the measurement of ideation quality. *Journal of Management Information Systems*, 23(4):143–161, 2007.

[65] Kim Renfro. For whom the troll trolls: A day in the life of a reddit moderator, Jan 2016.

[66] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36, 2017.

[67] Dan Siroker and Pete Koomen. *A/B testing: The most powerful way to turn clicks into customers*. John Wiley & Sons, 2013.

[68] Sky. Tokyo olympics: Transgender weightlifter laurel hubbard makes history - but fails to complete a lift, Aug 2021.

[69] Frans H van Eemeren, Bart Garssen, Erik CW Krabbe, A Francisca Snoeck Henkemans, Bart Verheij, and Jean HM Wagemans. Argumentation and artificial intelligence. In *Handbook of argumentation theory*, pages 1–51. Springer Netherlands, 2014.

[70] Frans H Van Eemeren, Rob Grootendorst, and Tjark Kruiger. *Handbook of argumentation theory*. De Gruyter Mouton, 2019.

[71] Bart Verheij. Dialectical argumentation with argumentation schemes: An approach to legal logic. *Artificial intelligence and Law*, 11(2):167–195, 2003.

[72] Serena Villata, Guido Boella, Dov M Gabbay, Leendert van der Torre, and Joris Hulstijn. A logic of argumentation for specification and verification of abstract argumentation frameworks. *Annals of Mathematics and Artificial Intelligence*, 66(1):199–230, 2012.

[73] Douglas Walton. Slippery slope arguments. *Journal of Applied Philosophy*, 1992.

[74] Douglas Walton. Justification of argumentation schemes. *The Australasian Journal of Logic*, 3, 2005.

[75] Douglas Walton. *Appeal to expert opinion: Arguments from authority*. Penn State Press, 2010.

[76] Douglas Walton. *Argumentation schemes for presumptive reasoning*. Routledge, 2013.

[77] Douglas Walton and David M Godden. The impact of argumentation on artificial intelligence. *Considering Pragma-Dialectics*, pages 287–299, 2006.

[78] Douglas Walton and Thomas F Gordon. Critical questions in computational models of legal argument. *Argumentation in artificial intelligence and law*, page 103, 2005.

[79] Douglas Walton, Christopher Reed, and Fabrizio Macagno. *Argumentation schemes*. Cambridge University Press, 2008.

[80] Etienne Wenger. *Communities of practice: Learning, meaning, and identity*. Cambridge university press, 1999.

[81] Adam Wyner and Trevor Bench-Capon. Argument schemes for legal case-based reasoning. In *JURIX*, pages 139–149. Citeseer, 2007.

Appendix A

Extra Information

A.1 Ethical clearance for UX testing

A.2 Gantt chart

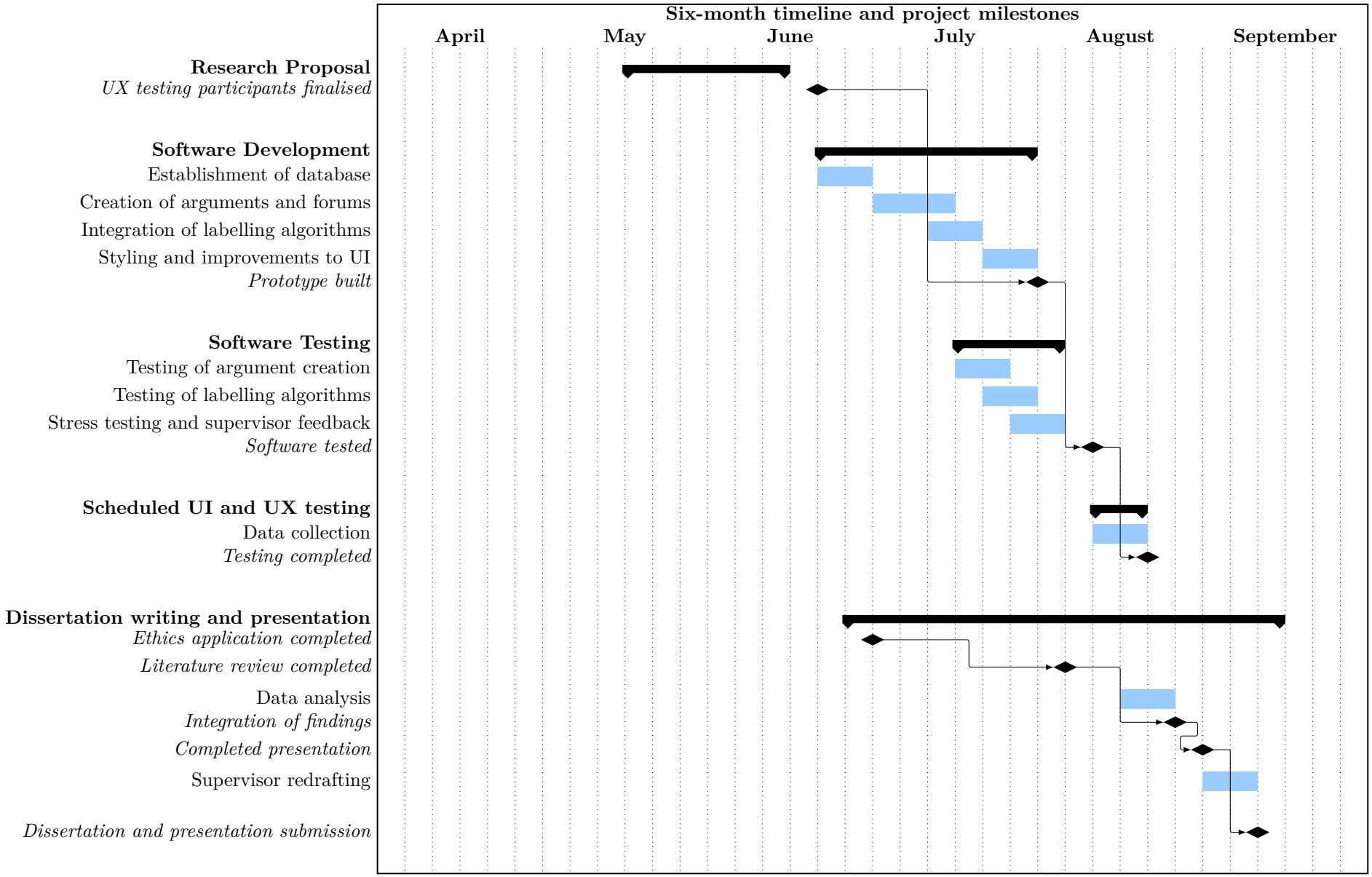
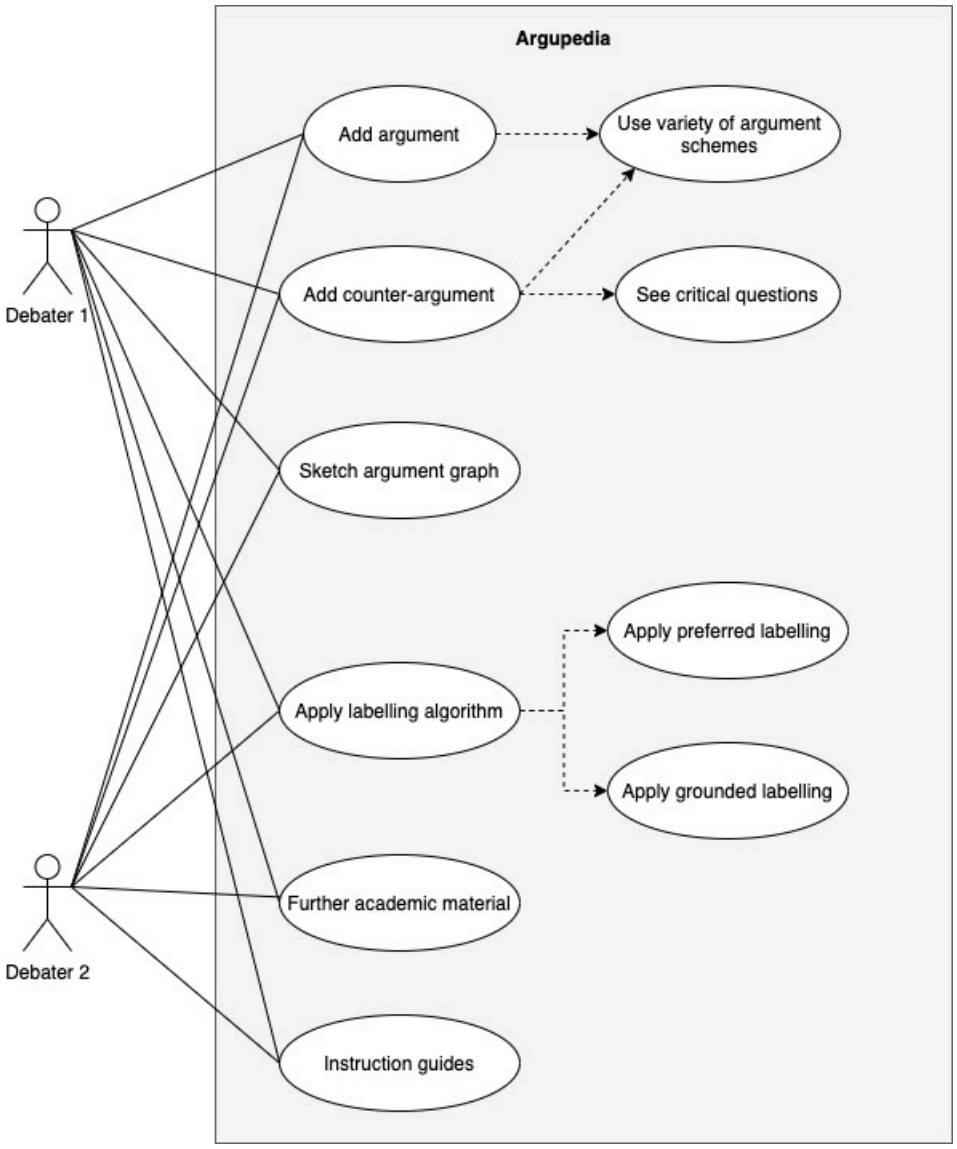
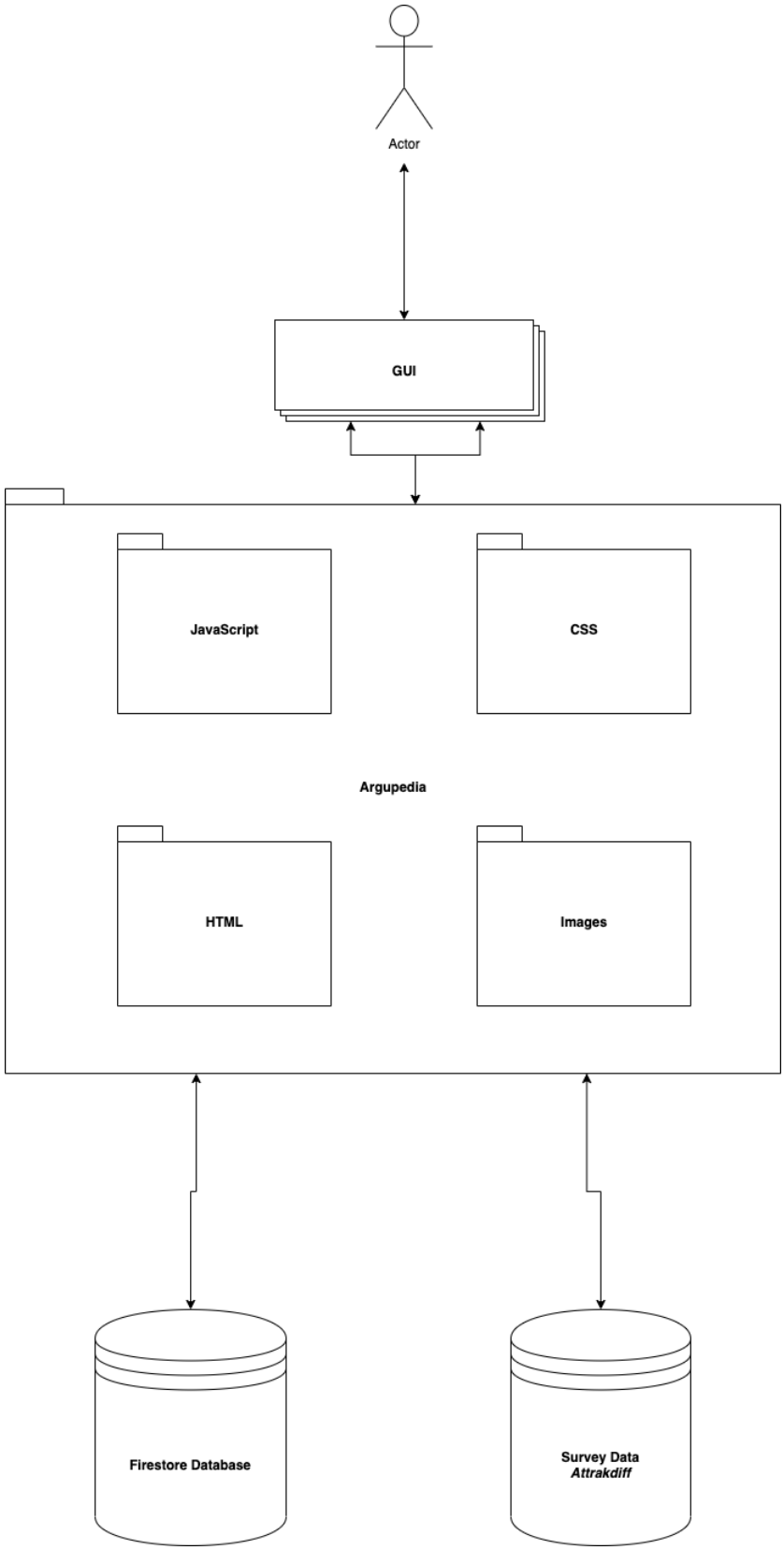


Figure A.1: Gantt chart of Argupedia proposed approach 2021. (Please note: blue bars denote progress made and diamonds milestones)

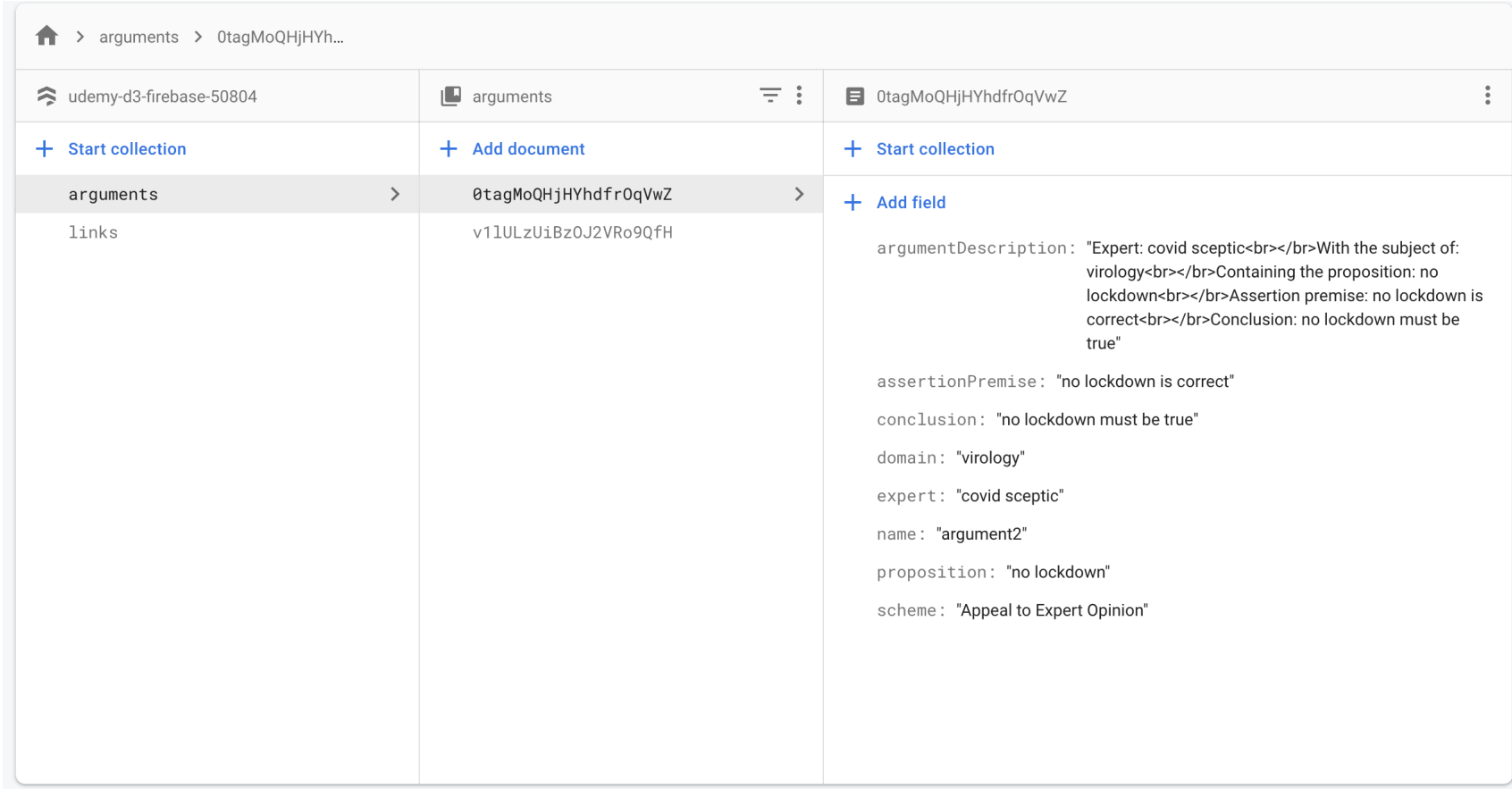
A.3 System use case diagram



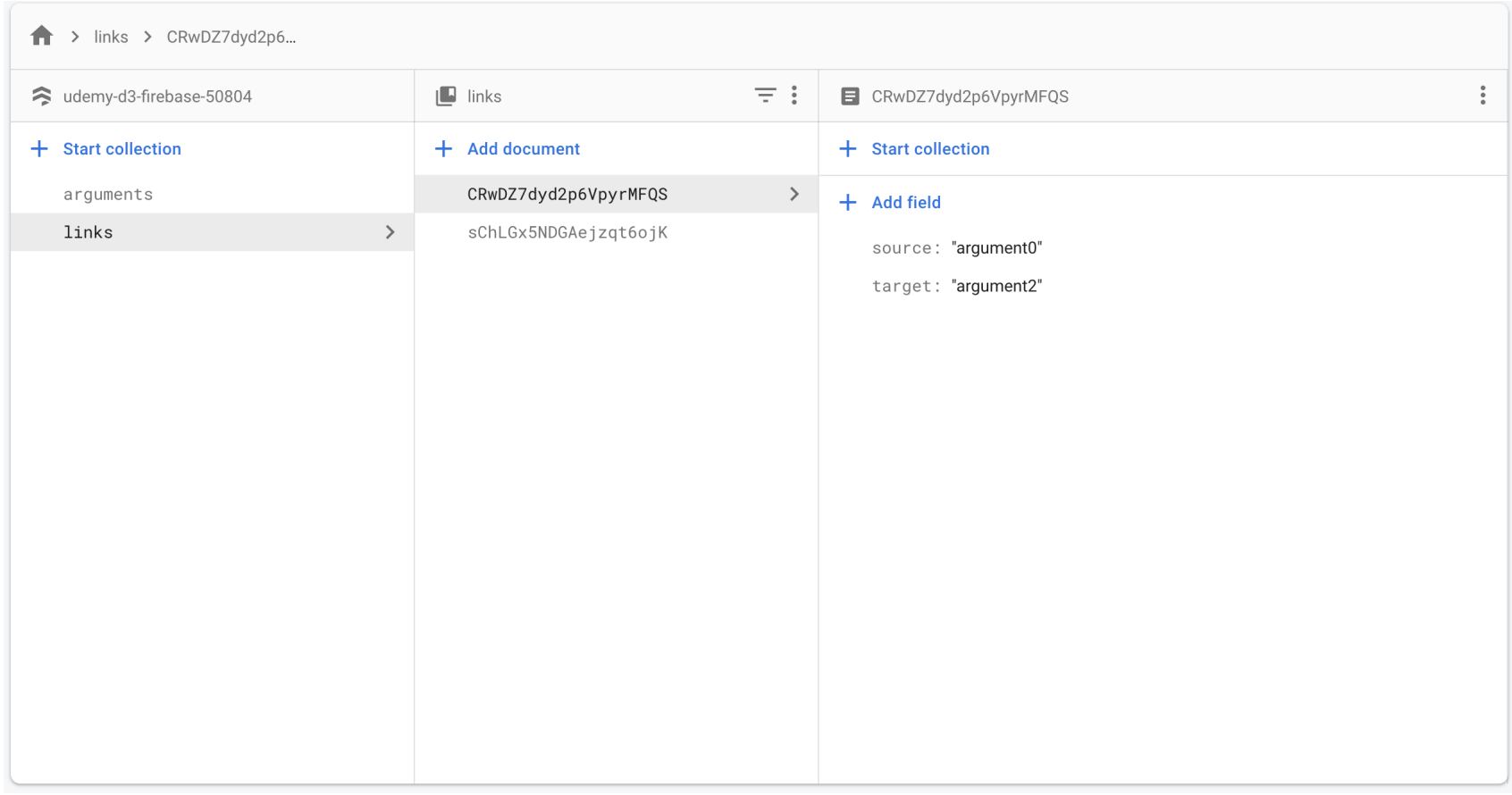
A.4 System package diagram



A.5 Argument database format example from Firestore



A.6 Links between attacking arguments database format example from Firestore

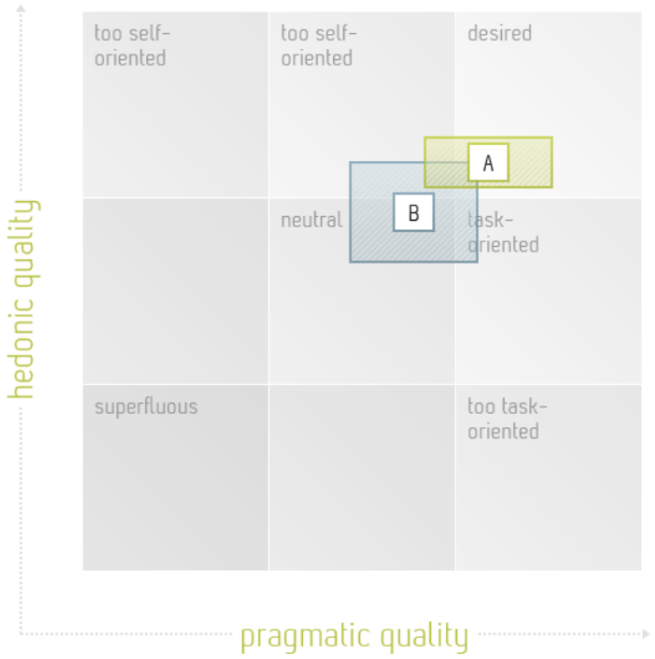


A.7 Attrakdif A/B UX testing instructions

Reference here: [30]

Comparison A-B

This type of study assesses each of the two different products separately and then compares them. You will be provided with an overview of how your customers perceive each of the products. You can decide whether both products are evaluated by the same test group, or whether product A and product B are evaluated by completely different test groups.



Each of the two products was evaluated by the 8 test participants (total of 8) using AttrakDiff.

The results were the following:

Product A performed better in both hedonic and pragmatic quality than product B. The smaller confidence rectangle observed for product A implies that users were largely at one. For product B however, the user responses differed greatly. In order that product B become competitive with product A it is necessary that improvements be made.

A.8 Argupedia black box testing results

Flow Graph Functionality Black Box Testing Report

Test Date: 01/08/2021
Test Description: Final Functionality Testing
Test Platform: MacOS Big Sur 11.4
Test Passed: 42/42
Test Failed: 0
Test Pass Rate: 100%

Key
Pass
Retest Pass
Fail

Test ID	Description	Test case	Expected result	Actual Result
1	Live server test	See if Argupedia runs and opens	No errors output	Pass
2	Use navbar to navigate	Navigate to different pages	Links between pages work	Pass
3	Navigate to YouTube video	Use inbuilt links to navigate to YouTube video	YouTube video works and plays successfully	Pass
4	Navigate and open surveys	Use navbar and inbuilt links to open survey	Surveys open and can input data	Pass
5	Find further argument scheme information	Use buttons to click for further information	Further information loads	Pass
6	Resize main window	Resize main window	Window resizes without distorting GUI	Pass
7	Find further Dung graph information	Use buttons to click for further information	Further information webpage loads	Pass
8	Initiate add initial argument	Click add initial argument button	Modal loads	Pass
9	Add initial argument with critical action scheme	Add initial argument, fill variables and see if graph updates	Add initial argument, fill variables and see if graph updates	Pass
10	Grounded labelling algorithm	Use switch	Graph node should be green and text of IN arguments appear	Pass

Test ID	Description	Test case	Expected result	Actual Result
11	Preferred labelling algorithm	Use switch	Graph node should be green and text of IN arguments appear	Pass
12	Complete labelling algorithm	Use switch	Graph node should be green and text of IN arguments appear	Pass
13	Add void initial argument with critical action scheme	Add initial argument, do not fill all variables	Argument should not lodge in repository successfully and error message appear	Pass
14	Resize main window	Resize main window	Window resizes without distorting GUI and graph	Pass
15	Add initial argument with appeal to expert opinion	Add initial argument, fill variables and see if graph updates	Add initial argument, fill variables and see if graph updates	Pass
16	Test on iPad sized screen	Resize main window	Window resizes without distorting GUI and graph	Pass
17	Add initial argument with appeal to popular opinion	Add initial argument, fill variables and see if graph updates	Add initial argument, fill variables and see if graph updates	Pass
18	Add initial argument with argument from analogy	Add initial argument, fill variables and see if graph updates	Add initial argument, fill variables and see if graph updates	Pass
19	Add initial argument with argument from consequences	Add initial argument, fill variables and see if graph updates	Add initial argument, fill variables and see if graph updates	Pass
20	Add initial argument with argument from correlation to cause	Add initial argument, fill variables and see if graph updates	Add initial argument, fill variables and see if graph updates	Pass
21	Add initial argument with argument from position to know	Add initial argument, fill variables and see if graph updates	Add initial argument, fill variables and see if graph updates	Pass
22	Add initial slippery slope argument	Add initial argument, fill variables and see if graph updates	Add initial argument, fill variables and see if graph updates	Pass
23	Test on iPhone sized screen	Resize main window	Window resizes without distorting GUI and graph	Pass
24	Create a counter-argument to argument0	Create a counter-argument without conflicting claims to argument0	Graph updates with link between arguments appropriately	Pass

Test ID	Description	Test case	Expected result	Actual Result
25	Grounded labelling algorithm	Use switch	All 8 arguments should update with 7 green and one red and text of IN arguments appear	**Pass**
26	Preferred labelling algorithm	Use switch	All 8 arguments should update with 7 green and one red and text of IN arguments appear	**Pass**
28	Complete labelling algorithm	Use switch	All 8 arguments should update with 7 green and one red and text of IN arguments appear	**Pass**
29	Create a counter-argument to argument1	Create a counter-argument with conflicting claims to argument0	Graph updates with bi-directional link between arguments appropriately	**Pass**
30	Press see critical questions in the process of creating counter-argument to critical action scheme	Use appropriate modal button	See that critical questions with dynamic variables have appeared	**Pass**
31	Grounded labelling algorithm	Use switch	All 8 arguments should update with 5 green, two yellow and one red and text of IN arguments appear Yellow should be labelled UNDEC	**Pass**
32	Preferred labelling algorithm	Use switch	All 8 arguments should update with 7 green and one red and text of IN arguments appear Conflicting claims should be IN/OUT labelled	**Pass**
33	Complete labelling algorithm	Use switch	All 8 arguments should update with 5 green, 2 yellow and 1 red and text of IN arguments appear Yellow should be labelled UNDEC, IN/OUT	**Pass**
34	Resize main window	Resize main window	Window resizes without distorting GUI and graph	**Pass**
35	Press see critical questions in the process of creating counter-argument to appeal to expert opinion	Use appropriate modal button	See that critical questions with dynamic variables have appeared	**Pass**

Test ID	Description	Test case	Expected result	Actual Result
36	Press see critical questions in the process of creating counter-argument to appeal to popular opinion	Use appropriate modal button	See that critical questions with dynamic variables have appeared	**Pass**
37	Press see critical questions in the process of creating counter-argument to argument from analogy	Use appropriate modal button	See that critical questions with dynamic variables have appeared	**Pass**
38	Press see critical questions in the process of creating counter-argument to argument from consequences	Use appropriate modal button	See that critical questions with dynamic variables have appeared	**Pass**
39	Press see critical questions in the process of creating counter-argument to argument from correlation to cause	Use appropriate modal button	See that critical questions with dynamic variables have appeared	**Pass**
40	Press see critical questions in the process of creating counter-argument to from position to know	Use appropriate modal button	See that critical questions with dynamic variables have appeared	**Pass**
41	Press see critical questions in the process of creating counter-argument to slippery slope	Use appropriate modal button	See that critical questions with dynamic variables have appeared	**Pass**
42	Clear argupedia	Delete all arguments from repository	Graph and interface update appropriately	**Pass**

Appendix B

User Guide

B.1 Instructions

B.1.1 YouTube video

For a full run-through of available commands and instructions - a video was developed for UX testing and sent to participants about Argupedia. That video is available here: Argupedia Tutorial.

B.1.2 Written instructions

About Argupedia

Argupedia is an online platform capable of enabling users to argue in a very different and profound way. Argupedia serves to prevent polarisation and offer users transparency over the arguments that are currently most preferred. In addition, Argupedia teaches users how to debate by offering an innovative GUI and rigorous argument schemes.

System requirements

The application is designed to run on any modern browser and was developed using the latest version of Google Chrome. To run code the application requires an OS capable of running JavaScript ES6 or higher.

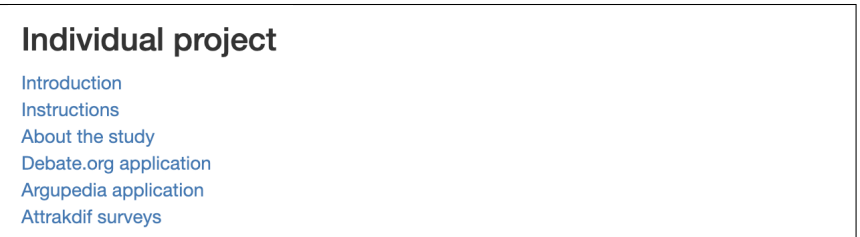
File structure

File structure is given by diagram in Appendix B.2.1.

Launching the application and deployment

The application can be launched by running the index.html in a live server through Google Chrome or other modern browser. Otherwise the application is hosted and deployed via my personal Github here: deployed Argupedia prototype.

Initial GUI controls

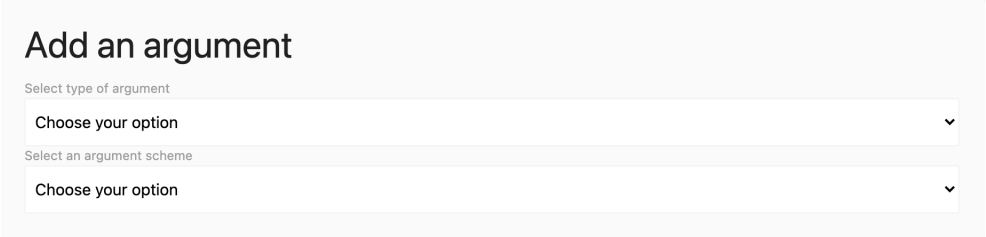
Main app navigation bar	
Argupedia navigation bar	
Add argument button	
Info framework button	
Info schemes button	

Navigation bars allow user to seamlessly navigate throughout web application. Whilst, the add argument button lets users add an argument to the argument state-space. Lastly, the information buttons provide background information concerning the academic underpinnings of Argupedia.

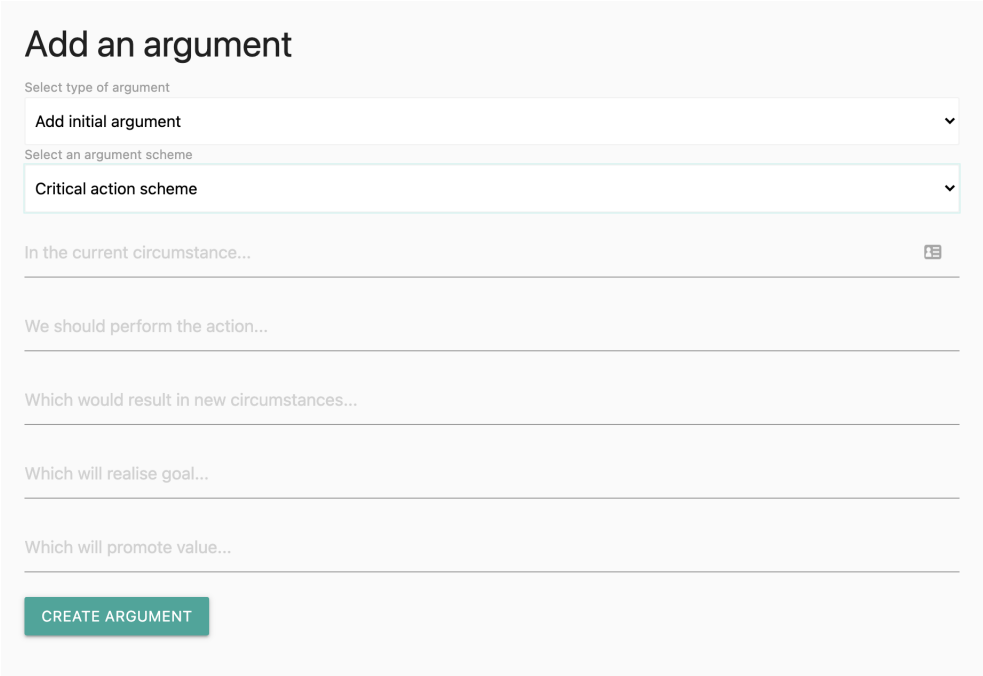
Adding an argument

The process of adding an argument is in bullet points to make the process especially clear.

- Users must first click the add argument button.
- Users are then presented with a modal and choices over type of argument and scheme.



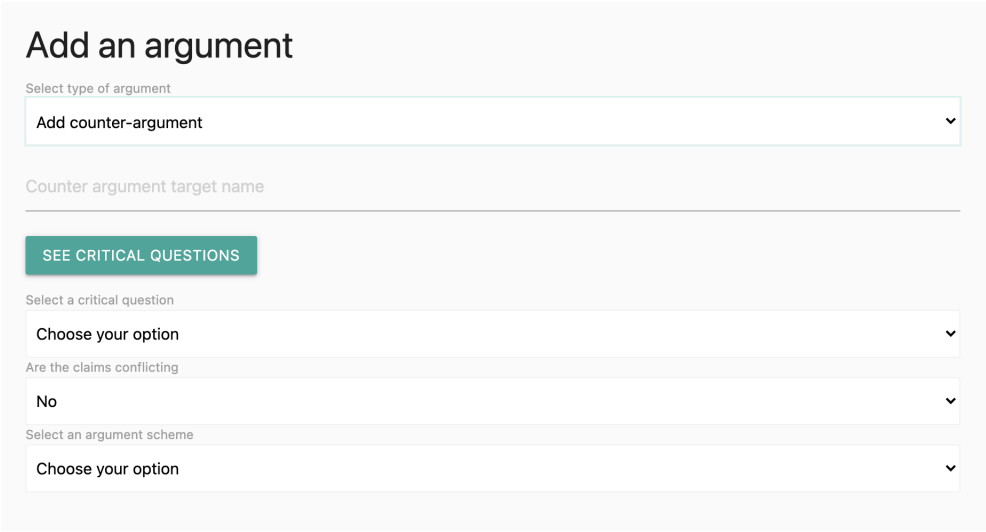
- Users then make choice of “initial argument” and argument scheme then the modal expands.



- Users must fill in variable and create initial argument

Adding a counter-argument without conflicting claims

- Users click the add argument button.
- Users are then presented with a modal and choices over type of argument and scheme.
- Users choose "add counter-argument" and modal expands.



- Users must fill in ID of argument they are arguing against - this is clearly given in state-space.
- Users click see critical questions button.
- Users must choose a critical question.
- Users leave conflicting claims as "No".

- Users choose argument scheme.
- Users must fill in variables and create counter-argument.
- Users must click create argument.

Adding an argument with conflicting claims

- Users must follow the same add counter-argument process above but choose "Yes" for conflicting claims.

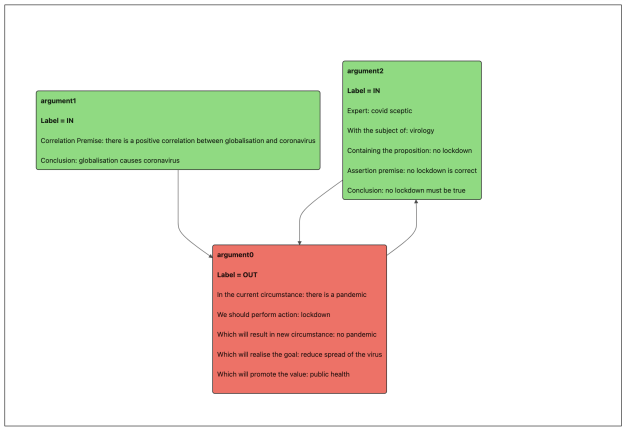
Manipulating the argumentation state space

Users can dynamically manipulate the argumentation state space and view the graph from any angle or zoom with either their mouse or track-pad or even fingers depending on device.

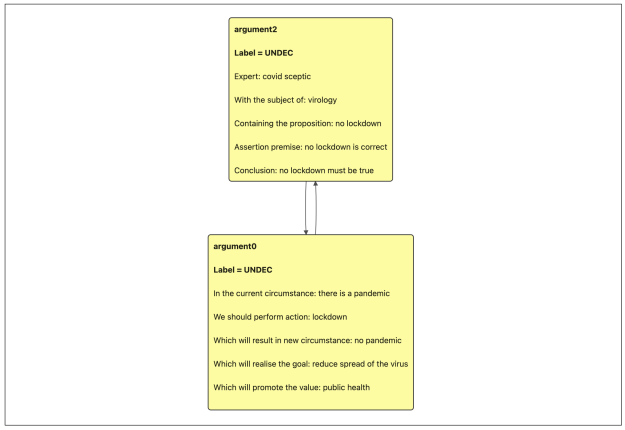
Applying labelling algorithms to the graph

- Switches are provided to adapt the graph with labelling algorithms.

-
- As displayed by the below figures Argupedia employs a traffic light system with the labelling algorithms. This means that filled green arguments are IN whilst OUT arguments are denoted with a red fill.



-
- In contrast, UNDEC arguments are provided with a yellow fill.



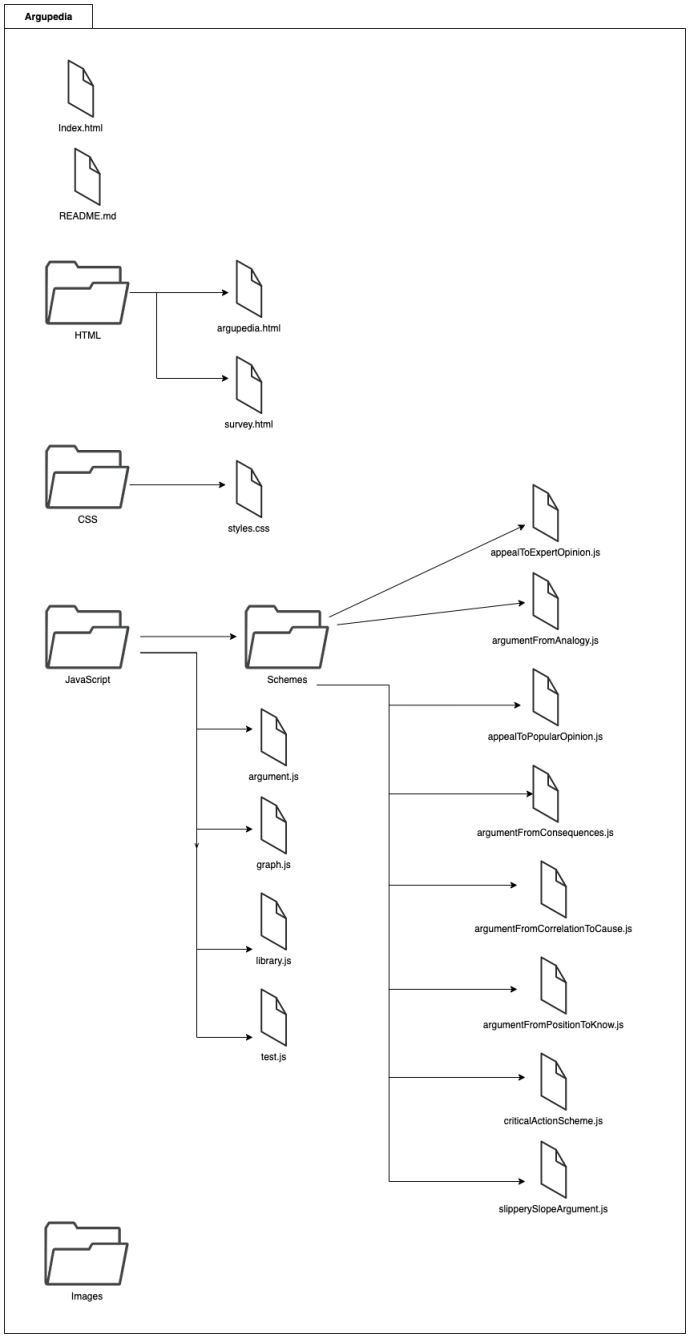
- ADD ARGUMENT

Filling out survey data

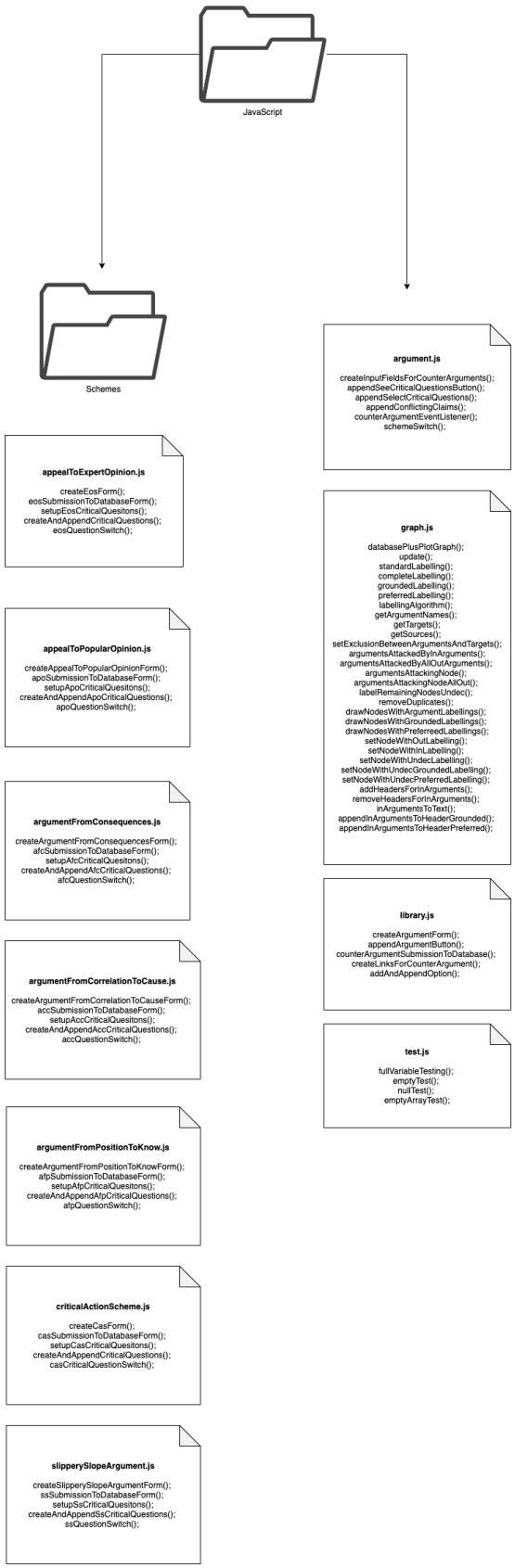
- Survey data can be easily filled out by users once they use the navigation bar to navigate to the correct page and are provided with instructions and links.

B.2 Diagrams

B.2.1 File structure



B.2.2 JavaScript folder functions



Appendix C

Source Code

C.1 Listings

I verify that I am the sole author of the programs contained in this folder, except where explicitly stated to the contrary.

anon, September 1, 2021.

C.2 Listings table

The following table documents the source code contained in the following pages. Source code files have been logically listed in the same order as the file structure diagram located in Appendix B.2.1. Each page of source code contains its page count and description.

Filename	Page count	Description
index.html	5	Main homepage file that ties all pages of the web application together.
README.md	1	Contains basic instructions.
html/argupedia.html	3	Argupedia web page file that initialises and configures firebase firestore database and is adapted with in-built switches and modal.
html/survey.html	2	Simple web-page host for survey links.
css/styles.css	1	File for styling of Argupedia application.
appealToExpertOpinion.js	4	Short javascript file to add appeal to expert opinion argument scheme.
appealToPopularOpinion.js	3	File to add appeal to popular opinion argument scheme.
argumentFromAnalogy.js	3	File for argument from analogy scheme.
argumentFromConsequences.js	4	File which adds argument from consequences scheme.
argumentFromCorrelationToCause.js	4	Short javascript file which adds argument from correlation to cause.
argumentFromPositionToKnow.js	4	Javascript file which adds argument from position to know.
argumentFromPositionToKnow.js	4	Javascript file which adds argument from position to know.
criticalActionScheme.js	5	Functionality for critical action scheme.
slipperySlopeArgument.js	4	Javascript file for slippery slope arguments.
argument.js	7	The back-end engine of adapting critical schemes, lodging arguments into the database and reading arguments from the database.
graph.js	13	The graphical engine to draw graphs using D3 and add requisite labelling algorithms.
library.js	3	A library file that can be imported for regularly used JavaScript functions.
test.js	2	A folder for testing of users inputted data and for unit testing.
images/	n/a	Generic images folder for icons and figures.

C.3 index.html

```
<!--
 *------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
---------Homepage File-----------------------------------------------------
------------------------
 -->
<!-- Source used for html page structure: https://hackr.io/blog/html-projects
-->
<html>
<!--This uses the default bootstrap stylesheet-->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
    integrity="sha384-
BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">
<!-- Provides a full-width container that can expand or collapse based on the
size of viewport-->

<head>
    <title>Homepage</title>
    <link rel="shortcut icon" type="image/x-icon" href="images/favicon.ico?
v=2" />
</head>

<body>
    <div class="container-fluid">
        <div class="row">
            <div class="col-md-2 col-sm-12 col-xs-12">
                <nav id="navbar">
                    <h3>Individual project</h3>
                    <!--Content stacking for smaller screens-->
                    <ul class="nav nav-pills nav-stacked">
                        <!--Internal linking to the respective sections-->
                        <a class="nav-link" href="index.html" rel="internal">
                            <li>Introduction</li>
                        </a>
                        <a class="nav-link" href="index.html" rel="internal">
                            <li>Instructions</li>
                        </a>
                        <a class="nav-link" href="index.html" rel="internal">
                            <li>About the study</li>
                        </a>
                        <a class="nav-link" href="https://www.debate.org/"
rel="external">
                            <li>Debate.org application</li>
                        </a>
```

```
42                          <a class="nav-link" href="html/argupedia.html"
rel="internal">
43                              <li>Argupedia application</li>
44                          </a>
45                          <a class="nav-link" href="html/survey.html"
rel="internal">
46                              <li>Attrakdif surveys</li>
47                          </a>
48                      </ul>
49                  </nav>
50              </div>
51              <div class="col-md-10 col-sm-12 col-xs-12">
52                  <h1>7CCSMPRJ Individual Project 2021</h1>
53                  <h3>King's College London</h3>
54                  <h5>Humphrey Curtis</h5>
55                  <p><a href="mailto:k20103881@kcl.ac.uk?subject=July Argupedia
User Study">k20103881@kcl.ac.uk</a></p>
56                  <main id="main-doc">
57                      <section class="main-section" id="Introduction">
58                          <h3 style="background: black; color:
white">Introduction</h3>
59                          <article>
60                              <p>Welcome to the Argupedia user study homepage,
thank you so much for taking the time to
61                                  check out the applications and provide your
critical thoughts and opinions! Use the
62                                  navbars to navigate freely around the
website.</p>
63                              <p>The core purpose of the study is to collect
your opinion via an anonymous survey on two
64                                  totally different
65                                  online debating applications - one the
largest public repository of online debates
66                                  called Debate.org and
67                                  the other a conceptual prototype based on
research within AI argumentation developed by
68                                  myself at
69                                  King's College London for my summer MSc
Advanced Computer Science thesis. Ultimately the
70                                  user study should be fun and interesting and
you are successfully contributing towards
71                                  critical research on the development of
future web applications!</p>
72                              <code>
73                                  Please be sure to spend some time reading all
below instructions and watch the
74                                  respective YouTube videos before proceeding
to testing and evaluating either debating
75                                  platforms and filling out the surveys.
76                              </code>
77                              <p></p>
78                              <p>If you have any questions, thoughts or
concerns at any point please do not hesitate to
79                                  use the above link to send me an email.</p>
80                          </article>
81                      </section>
82                      <section class="main-section" id="Instructions">
83                          <h3 style="background: black; color:
white">Instructions</h3>
84                          <article>
```

```
                            <p>The user study is (hopefully) super simple and
should only take 45 minutes to an hour at
                                most. Read all
                                the three sets of bullet-point instructions
and
                                follow them in order - ultimately a laptop or
desktop computer is suitable to best
                                experience both platforms, however if you
choose to use a mobile or tablet, that is
                                totally fine too!</p>
                            <code>Without further ado, turning to the bullet
point instructions for Debate.org:</code>
                            <p></p>
                            <li>First, use the navbar link to navigate to the
worlds largest online debate platform
                                called
                                Debate.org.</li>
                            <li>Spend some time on Debate.org familiarising
yourself with the website and platform.</li>
                            <li>Secondly, try to spend some time interacting
with Debate.org.</li>
                            <li>Interaction can mean reading some active
debates, creating a Debate.org account and
                                contributing to live debates, starting new
debates, contributing to forums and polls.
                            </li>
                            <li>Debate.org is open source and does not
require an account - you can easily contribute
                                anonymously.</li>
                            <li>If you are already an expereinced member of
Debate.org that is totally fine and you can
                                proceed to testing the Argupedia application.
</li>
                            <li>Finally, after 20 to 45 minutes spent on
Debate.org click on the Attrakdif
                                survey link in the
                                Navbar and navigate to filling out a survey
on your experience and opinions of the
                                Debate.org platform.</li>
                            <p></p>
                            <code>Instructions for the Attrakdif survey:
</code>
                            <p></p>
                            <li>Use the navbar to reach the Attrakdif survey
links.</li>
                            <li>Click on the correct link for either the
Debate.org or the Argupedia multiple choice
                                survey.</li>
                            <li>An Attrakdif survey should take roughly 3 to
5 minutes to complete.</li>
                            <li>The Attrakdif survey uses word pairs in which
you rate the platform between two starkly
                                contrasted words.</li>
                            <li>Do not ponder over the word pairs and make
your assessment spontaneously - you may feel
                                that some word pairs do not fit the product
very well.</li>
                            <li>However, I would ask you to give an answer
anyway. Remember at all times that there are
```

```
                                no "right" or "wrong" answers - your personal opinion is what counts.</li>
                            <p></p>
                            <code>Lastly, instructions for the Argupedia platform:</code>
                            <p></p>
                            <li>Firstly, please watch the following YouTube video will hopefully make the intearction
                                with the Argupedia platform totally clear. </li>
                            <li>Please click here for a video on: <a href="https://youtu.be/1BaNoLEGmrU">Argupedia instructions</a></li>
                            <li>Secondly, don't forget to use the navbar links to fill out the Attrakdif survey for
                                Argupedia - it will be the exact same survey again!</li>
                            <p></p>
                            <code>Congratulations - you're all done give yourself a well-deserved pat on the
                                back!</code>
                            <p></p>
                        </article>
                    </section>
                    <section class="main-section" id="aboutStudy">
                        <h3 style="background: black; color: white">About the study</h3>
                        <article>
                            <p>This study consists of <a href="https://en.wikipedia.org/wiki/A/B_testing">performing A/B
                                    testing</a> of two debating applications (click the link for more information from
                                Wikipedia). As noted
                                earlier Debate.org is the current largest public debating platform online. Whilst, in
                                stark contrast
                                Argupedia is a prototype application based off the latest advances in AI argumentation
                                research for my thesis
                                project at King's College London advised by Professor Sanjay Modgil.
                            </p>
                            <p> The purpose of the study is to gain initial feedback from participants on which of the
                                two applications
                                they find most effective, engaging and interactive. Participants initial feedback will then
                                be harnessed and used to
                                improve the Argupedia application for future research. Furthermore, if participants are
                                willing to take an informal interview or share any helpful thoughts
                                - please do not hesitate to send me an email at your own discretion using the above
                                email link. Many thanks for your contribution!</p>
                            <p></p>
                        </article>
                    </section>
                    <section class="main-section" id="researchEthics">
```

```
                        <h3 style="background: black; color: white">Research ethics and further information</h3>
                        <article>
                            <p>This study recieved ethical clearance from King's College London - it is classed as
                                minimal ethical risk. My details can be used at any point if you have questions,
                                complaints or
                                withdrawal requests. Participants have the right to withdraw either during or after
                                participation without having to provide a reaosn. All data collected from the surveys is
                                deliberately totally
                                anonymous and there is no way in which personal identifying details can be collected via
                                the survey. By
                                submitting a multiple choice Attrakdif survey you are providing informed consent that I
                                may use your survey results as a part of my thesis - the survey results will be
                                presented fully anonymously. The Attrakdif survey data
                                results will be securely stored in line with the Data Protection Act (1998) and King's
                                College London guidance.
                            </p>
                        </article>
                    </section>
                </main>
            </div>
        </div>
    </div>

    <img src="images/kingsLogo.png"
        style="width: 320px; height: 290px; margin-left: auto; margin-right: auto; display: block; padding-top: 50px; padding-bottom: 10px;"
        alt="Kings Logo">
</body>

</html>
```

C.4 html/argupedia.html

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <!-- Compiled and minified CSS -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0-
rc.2/css/materialize.min.css">
    <!-- Materialise CSS -->
    <link rel="stylesheet" href="../css/styles.css">
    <!-- Import Google icon fonts  -->
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
    <title>Argupedia</title>
    <link rel="shortcut icon" type="image/x-icon"
href="../images/favicon.ico?v=2" />
</head>

<body>
    <nav>
        <div class="nav-wrapper">
            <a href="#" class="brand-logo right">Argupedia</a>
            <ul id="nav-mobile" class="left hide-on-med-and-down">
                <li><a href="../index.html">Introduction</a></li>
                <li><a href="../index.html">Instructions</a></li>
                <li><a href="../index.html">About the study</a></li>
                <li><a href="https://www.debate.org/"
rel="external">Debate.org</a></li>
                <li><a href="survey.html">Attrakdif survey</a></li>
            </ul>
        </div>
    </nav>

    <div class="grey lighten-3 section grey-text" style="position: relative;
margin-bottom: 10px">
        <p class="flow-text center">The number ONE AI argumentation platform!
</p>
    </div>

    <div>
        <a class="btn waves-effect waves-light red right" style="margin:
15px"

 href="https://philosophicaldisquisitions.blogspot.com/2010/03/argumentation-
schemes-part-1.html">Schemes explained</a>
        <a class="btn waves-effect waves-light red right" style="margin:
15px"

 href="https://en.wikipedia.org/wiki/Argumentation_framework">Intro to Dung
framework</a>
    </div>

    <div class="container">
        <div class="canvas">
            <svg id="svg" style="border:1px solid black"></svg>
        </div>
    </div>

```

```
    <!-- Second modal button for counter-arguments -->
    <a class="waves-effect waves-light btn modal-trigger" href="#modal1" style="margin: 30px">Add argument</a>

    <!-- Modal Structure -->
    <div id="modal1" class="modal">
        <div class="modal-content">
            <h4>Add an argument</h4>
            <label>Select type of argument</label>
            <select class="browser-default" id="selectArgument">
                <option value="reset" disabled selected>Choose your option</option>
                <option value="initialArgument">Add initial argument</option>
                <option value="counterArgument">Add counter-argument</option>
            </select>
            <div id="counterArgumentInputFields"></div>
            <label>Select an argument scheme</label>
            <select class="browser-default" id="selectArgumentScheme2">
                <option value="reset" disabled selected>Choose your option</option>
                <option value="casForm">Critical action scheme</option>
                <option value="appealToExpertOpinion">Appeal to Expert Opinion</option>
                <option value="appealToPopularOpinionForm">Appeal to Popular Opinion</option>
                <option value="argumentFromAnalogy">Argument from Analogy</option>
                <option value="argumentFromCorrelationToCause">Argument from Correlation to Cause</option>
                <option value="argumentFromConsequences">Argument from Consequences</option>
                <option value="slipperySlopeArgument">Slippery Slope Argument</option>
                <option value="argumentFromPositionToKnow">Argument from the Position To Know</option>
            </select>
            <form class="hide" id="counterArgumentScheme">
            </form>
        </div>
    </div>



    <div class="switch" id="switchDiv">
        <h6>Grounded labelling</h6>
        <label>
            No
            <input type="checkbox" id="groundedSwitch">
            <span class="lever"></span>
            Yes
        </label>
        <br></br>
        <h6>Preferred labelling</h6>
        <label>
            No
            <input type="checkbox" id="preferredSwitch">
            <span class="lever"></span>
            Yes
        </label>
        <br></br>
```

```
        <h6>Complete labelling</h6>
        <label>
            No
            <input type="checkbox" id="mySwitch">
            <span class="lever"></span>
            Yes
        </label>
        <br></br>
    </div>

    <!-- The core Firebase JS SDK is always required and must be listed first -->
    <script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-app.js"></script>
    <script src="https://www.gstatic.com/firebasejs/8.6.8/firebase-firestore.js"></script>


    <script>
        // Your web app's Firebase configuration
        // For Firebase JS SDK v7.20.0 and later, measurementId is optional
        var firebaseConfig = {
            apiKey: "AIzaSyAHCuY34yoA67g3eCUQf4z8m-S_nEz__bc",
            authDomain: "udemy-d3-firebase-50804.firebaseapp.com",
            projectId: "udemy-d3-firebase-50804",
            storageBucket: "udemy-d3-firebase-50804.appspot.com",
            messagingSenderId: "252447511874",
            appId: "1:252447511874:web:7e14f2ffee0df61e2b64db",
            measurementId: "G-JK3NTPXHF6"
        };
        // Initialize Firebase
        firebase.initializeApp(firebaseConfig);
        //   firebase.analytics();
        const db = firebase.firestore();
        const settings = {
            timestampsInSnapshots: true,
            merge: true
        };
        db.settings(settings);
    </script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0-rc.2/js/materialize.min.js"></script>
    <script src="https://d3js.org/d3.v4.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/dagre-d3/0.6.3/dagre-d3.js"></script>
    <script type="module" src="../javascript/argument.js"></script>
    <script src="../javascript/graph.js"></script>
</body>

</html>
```

C.5 html/survey.html

```
<!--
 *-----------------------------------------------------------------------
-----------------------------------------------------------------------
----------------------
 * -----------------------------------------------------------------------
-----------------------------------------------------------------------
----------------------
 * -----------------------------------------------------------------------
-----------------------------------------------------------------------
----------------------
 * -----------------------------------------------------------------------
-----------------------------------------------------------------------
----------------------
 * -----------------------------------------------------------------------
-----------------------------------------------------------------------
----------------------
 * -----------------------------------------------------------------------
--------Survey links page-----------------------------------------------
----------------------
-->
<!-- Source used for html page structure: https://hackr.io/blog/html-projects
-->
<html>
<!--This uses the default bootstrap stylesheet-->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"
    integrity="sha384-
BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">
<!-- Provides a full-width container that can expand or collapse based on the
size of viewport-->

<head>
    <title>Attrakdif surveys</title>
    <link rel="shortcut icon" type="image/x-icon" href="../images/favicon.ico?
v=2>
</head>

<body>
    <nav id="navbar">
        <div class="container-fluid">
            <div class="row">
                <div class="col-md-2 col-sm-12 col-xs-12">
                    <nav id="navbar">
                        <h3>Individual project</h3>
                        <!--content stacking for smaller screens-->
                        <ul class="nav nav-pills nav-stacked">
                            <!--internal linking to the respective sections-->
                            <a class="nav-link" href="../index.html"
rel="internal">
                                <li>Introduction</li>
                            </a>
                            <a class="nav-link" href="../index.html"
rel="internal">
                                <li>Instructions</li>
                            </a>
                            <a class="nav-link" href="../index.html"
rel="internal">
                                <li>About the study</li>
                            </a>
```

```
                            <a class="nav-link" href="https://www.debate.org/"
rel="external">
                                <li>Debate.org application</li>
                            </a>
                            <a class="nav-link" href="argupedia.html"
rel="internal">
                                <li>Argupedia application</li>
                            </a>
                            <a class="nav-link" href="survey.html"
rel="internal">
                                <li>Attrakdif surveys</li>
                            </a>
                        </ul>
                    </nav>
                </div>

                <div class="col-md-10 col-sm-12 col-xs-12">
                    <h1>Attrakdif Surveys</h1>
                    <h3>King's College London</h3>
                    <h5>Humphrey Curtis</h5>
                    <p><a href="mailto:k20103881@kcl.ac.uk?subject=July
Argupedia User Study">k20103881@kcl.ac.uk</a>
                    </p>
                    <main id="main-doc">
                        <section class="main-section" id="DebateSurvey">
                            <h3 style="background: black; color:
white">Debate.org Survey</h3>
                            <li>Please use the following link (click to open
in a new tab): <a

 href="https://esurvey.uid.com/survey/#230d6a33-fe4c-4d17-b9b8-
0724f47dc939">for the
                                    Debate.org survey click here</a></li>
                        </section>
                        <main id="main-doc">
                            <section class="main-section" id="DebateSurvey">
                                <h3 style="background: black; color:
white">Argupedia Survey</h3>
                                <li>Please use the following link (click to
open in a new tab): <a

 href="https://esurvey.uid.com/survey/#cf9e6186-b8ca-49d2-9a72-
ba36fd806f88">for
                                        the Argupedia survey link click
here</a></li>
                            </section>
                </div>
            </div>
        </div>
    </nav>

</body>
```

C.6 css/styles.css

```css
body {
    margin: 0;
    top: 0;
    right: 0;
    bottom: 0;
    left: 0;
}

text {
    font-weight: 100;
    font-family: "Helvetica Neue", Helvetica, Arial, sans-serf;
    font-size: 5px;
}

.node rect {
    stroke: #333;
    fill: #fff;
    stroke-width: 1.5px;
}

.edgePath path.path {
    stroke: #333;
    fill: none;
    stroke-width: 1.5px;
    opacity: 0.7
}

.arrowhead {
    stroke: blue;
    fill: blue;
    stroke-width: 1.5px;
}

.switch {
    margin-bottom: 30px;
    margin-left: 30px;
}


svg {
    width: calc(100% - 20px);
    height: calc(100% - 150px);
}
```

C.7 javascript/schemes/appealToExpertOpinion.js

```
1  import * as lib from "../library.js";
2  import * as test from "../test.js";
3
4  /* --------------------------------------------------------------------------
   ------------------------------------------------------------------------
   -------------------------
5   * --------------------------------------------------------------------------
   ------------------------------------------------------------------------
   -------------------------
6   * --------------------------------------------------------------------------
   ------------------------------------------------------------------------
   -------------------------
7   * --------------------------------------------------------------------------
   ------------------------------------------------------------------------
   -------------------------
8   * --------------------------------------------------------------------------
   ------------------------------------------------------------------------
   -------------------------
9   * ------------------------------------------------------------------
   Argumentation functionality ----------------------------------------------
   -------------------------
10  */
11
12 var numberOfArguments = 0;
13
14 const selectTypeOfArgument = document.getElementById("selectArgument");
15 const selectArgumentScheme =
   document.getElementById("selectArgumentScheme2");
16 const argumentForm = document.getElementById("counterArgumentScheme");
17
18 /*
19  * Function which updates variable status of number of arguments to create
   unique ID
20  *
21  */
22 db.collection("arguments").onSnapshot(function (querySnapshot) {
23
24     // console.log("Number of arguments = " + querySnapshot.docs.length);
25     numberOfArguments = querySnapshot.docs.length;
26 });
27
28 /*
29  * Function which creates appeal to expert opinion form and has event
   listener for submission of form
30  * Also recognises if user is submitting counter-attacking argument
31  *
32  */
33 var createEosForm = (function (elementToAppend, id, buttonClass, buttonId,
   modalName, argumentStatus) {
34
35     lib.createArgumentForm(elementToAppend, "Expert E...", "eosExpert");
36     lib.createArgumentForm(elementToAppend, "Is an expert in domain D...",
   "eosDomain");
37     lib.createArgumentForm(elementToAppend, "Containing proposition A...",
   "eosProposition");
38     lib.createArgumentForm(elementToAppend, "E asserts that A (in domain D)
   is true (or false)...", "eosAssertionPremise");
39     lib.createArgumentForm(elementToAppend, "A may be plausibly taken to be
   true", "eosConclusion");
40
```

```
    lib.appendArgumentButton(elementToAppend, buttonClass, buttonId);
    var argumentSubmissionButton = document.getElementById(buttonId);

    argumentSubmissionButton.addEventListener("click", function () {
        eosSubmissionToDatabaseForm(id, modalName);

        if (argumentStatus == "counterArgument") {
            lib.counterArgumentSubmissionToDatabase(numberOfArguments);
        }
    });
});

/*
 * Function which performs submission to database of user inputted data
 *
 */
var eosSubmissionToDatabaseForm = (function (id, modalName) {
    var form = document.getElementById(id);

    var eosExpert = document.querySelector("#eosExpert");
    var eosDomain = document.querySelector("#eosDomain");
    var eosProposition = document.querySelector("#eosProposition");
    var eosAssertionPremise = document.querySelector("#eosAssertionPremise");
    var eosConclusion = document.querySelector("#eosConclusion");


    var variables = [];
    variables.push(eosExpert.value, eosDomain.value, eosProposition.value,
eosAssertionPremise.value, eosConclusion.value);
    test.fullVariableTesting(variables);

    // var argumentFromUser = eosExpert.value + " -> " + eosDomain.value + "
-> " + eosProposition.value + " -> " + eosAssertionPremise.value + " -> " +
eosConclusion.value;
    var argumentFromUser = "Expert: " + eosExpert.value.toLowerCase() +
    "<br></br>With the subject of: " + eosDomain.value.toLowerCase() +
    "<br></br>Containing the proposition: " +
eosProposition.value.toLowerCase() +
    "<br></br>Assertion premise: " + eosAssertionPremise.value.toLowerCase()
+
    "<br></br>Conclusion: " + eosConclusion.value.toLowerCase();

    // console.log("Argument = " + argumentFromUser);

    db.collection("arguments").add({
        name: "argument" + numberOfArguments,
        scheme: "Appeal to Expert Opinion",
        argumentDescription: argumentFromUser,
        expert: eosExpert.value.toLowerCase(),
        domain: eosDomain.value.toLowerCase(),
        proposition: eosProposition.value.toLowerCase(),
        assertionPremise: eosAssertionPremise.value.toLowerCase(),
        conclusion: eosConclusion.value.toLowerCase()
    });

    /*----- Reset modal fields after argument submission ----*/
    var instance = M.Modal.getInstance(modalName);
    instance.close();

    selectTypeOfArgument.selectedIndex = "reset";
```

```
    selectTypeOfArgument.disabled = false;
    selectArgumentScheme.selectedIndex = "reset";
    selectArgumentScheme.disabled = false;
    argumentForm.className = "hide";

    form.reset();

    eosExpert.remove();
    eosDomain.remove();
    eosProposition.remove();
    eosAssertionPremise.remove();
    eosConclusion.remove();

    var argumentButton = document.getElementById("counterArgumentButton");
    argumentButton.remove();
});

/* ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------------------
Critical question functionality ----------------------------------------
----------------------------
 */

/*
 * Function which dynamically builds critical questions
 *
 */
var setupEosCriticalQuestions = (function (data) {
    var expert;
    var domain;
    var proposition;
    var assertionPremise;
    var conclusion;

    data.forEach(function(d){
        expert = d.expert;
        domain = d.domain;
        proposition = d.proposition;
        assertionPremise = d.assertionPremise;
        conclusion = d.conclusion;
    });

    createAndAppendEosCriticalQuestions(expert, domain, proposition,
assertionPremise, conclusion);
})

```

```
var createAndAppendEosCriticalQuestions = (function(expert, domain, proposition, assertionPremise, conclusion){
    for (let i=1; i<7; i++){
        var tempCriticalQuestion = eosQuestionSwitch(i, expert, domain, proposition, assertionPremise, conclusion);
        lib.addAndAppendOption(tempCriticalQuestion);
    }
});

/*
 * E - expert
 * D - domain
 * A - proposition
 *
 */
var eosQuestionSwitch = (function(questionNumber, expert, domain, proposition, assertionPremise, conclusion) {
    switch(questionNumber) {
        case 1:
            return "How credible is \"" + expert + "\" as an expert?";
            break;
        case 2:
            return "Is \"" + expert +  "\" actually an expert in the field that \"" + proposition + "\" is in?";
            break;
        case 3:
            return "What did \"" + expert + "\" assert that implies \"" + proposition + "\"?";
            break;
        case 4:
            return "Is \"" + expert + "\" personally reliable and trustworthy? Do we have any reason to think that \"" + expert + "\" is less than honest?";
            break;
        case 5:
            return "Is \"" + proposition + "\" consistent with what other experts have asserted?";
            break;
        case 6:
            return "Is the evidence provided by \"" + expert + "\" based on actual evidence?";
            break;
        default:
            "Select a scheme to generate a critical question";
    }
});

export {
    createEosForm,
    setupEosCriticalQuestions
};

```

C.8 javascript/schemes/appealToPopularOpinion.js

```
1  import * as lib from "../library.js";
2  import * as test from "../test.js";
3
4  /* --------------------------------------------------------------------------------------------------------------
5   * --------------------------------------------------------------------------------------------------------------
6   * --------------------------------------------------------------------------------------------------------------
7   * --------------------------------------------------------------------------------------------------------------
8   * --------------------------------------------------------------------------------------------------------------
9   * ------------------------------------------------------------------ Argumentation functionality ---------------
10  */
11
12 var numberOfArguments = 0;
13
14 const selectTypeOfArgument = document.getElementById("selectArgument");
15 const selectArgumentScheme = document.getElementById("selectArgumentScheme2");
16 const argumentForm = document.getElementById("counterArgumentScheme");
17
18 /*
19  * Function which updates variable status of number of arguments to create unique ID
20  *
21  */
22 db.collection("arguments").onSnapshot(function (querySnapshot) {
23
24     // console.log("Number of arguments = " + querySnapshot.docs.length);
25     numberOfArguments = querySnapshot.docs.length;
26
27 });
28
29 /*
30  * Function which creates appeal to popular opinion form and has event listener for submission of form
31  * Also recognises if user is submitting counter-attacking argument
32  *
33  */
34 var createAppealToPopularOpinionForm = (function (elementToAppend, id, buttonClass, buttonId, modalName, argumentStatus) {
35     lib.createArgumentForm(elementToAppend, "A is generally accepted as being true...", "apoGeneralAcceptancePremise");
36     lib.createArgumentForm(elementToAppend, "If A is generaly accepted as being true that gives a reason in favour of A...", "apoPresumptionPremise");
37     lib.createArgumentForm(elementToAppend, "There is a reason in favour of A...", "apoOpinion");
38
39     lib.appendArgumentButton(elementToAppend, buttonClass, buttonId);
40     var argumentSubmissionButton = document.getElementById(buttonId);
41
```

```
    argumentSubmissionButton.addEventListener("click", function () {
        apoSubmissionToDatabaseFromForm(id, modalName);

        if (argumentStatus == "counterArgument") {
            lib.counterArgumentSubmissionToDatabase(numberOfArguments);
        }
    });
});

/*
 * Function which performs submission to database of user inputted data
 *
 */
var apoSubmissionToDatabaseFromForm = (function (id, modalName) {
    var form = document.getElementById(id);

    var premise = document.querySelector("#apoGeneralAcceptancePremise");
    var presumptionPremise =
document.querySelector("#apoPresumptionPremise");
    var opinion = document.querySelector("#apoOpinion");

    var variables = [];
    variables.push(premise.value, presumptionPremise.value, opinion.value);
    test.fullVariableTesting(variables);

    var argumentFromUser = "General acceptance premise: " +
premise.value.toLowerCase() +
    "<br></br>Presumption Premise: " + presumptionPremise.value.toLowerCase()
+
    "<br></br>Conclusion: " + opinion.value.toLowerCase();

    db.collection("arguments").add({
        name: "argument" + numberOfArguments,
        scheme: "Appeal to Popular Opinion",
        argumentDescription: argumentFromUser,
        userPremise: premise.value.toLowerCase()
    });

    var instance = M.Modal.getInstance(modalName);
    instance.close();

    selectTypeOfArgument.selectedIndex = "reset";
    selectTypeOfArgument.disabled = false;
    selectArgumentScheme.selectedIndex = "reset";
    selectArgumentScheme.disabled = false;
    argumentForm.className = "hide";

    form.reset();

    premise.remove();
    presumptionPremise.remove();
    opinion.remove();

    var argumentButton = document.getElementById("counterArgumentButton");
    argumentButton.remove();
});

/* ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
```

```
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------
Critical question functionality ------------------------------------------
------------------------------
 */

/*
 * Function which dynamically builds critical questions
 *
 */
var setupApoCriticalQuestions = (function (data) {
    var premise;

    data.forEach(function (d) {
        premise = d.userPremise;
    });
    // console.log("apo data = ", data);
    // console.log("premise = ", premise);

    createAndAppendApoCriticalQUestions(premise);
});

var createAndAppendApoCriticalQUestions = (function (premise) {
    for (let i = 1; i < 3; i++) {
        var tempCriticalQuestion = apoCriticalQuestionSwitch(i, premise);
        lib.addAndAppendOption(tempCriticalQuestion, i);
    }

});

var apoCriticalQuestionSwitch = (function (questionNumber, premise) {
    switch (questionNumber) {
        case 1:
            return "What evidence do we have for believing that \"" + premise
+ "\" is generally accepted?";
            break;
        case 2:
            return "Even if \"" + premise + "\" is generally accepted as
being true, are there good reasons for doubting its veracity?";
            break;
        default:
            "Select a scheme to genearte critical questions";
    }
});

export {
    createAppealToPopularOpinionForm,
    setupApoCriticalQuestions
};
```

C.9 javascript/schemes/argumentFromConsequences.js

```
import * as lib from "../library.js";
import * as test from "../test.js";

/* ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------
Argumentation functionality ----------------------------------------------
----------------------------
 */

var numberOfArguments = 0;

const selectTypeOfArgument = document.getElementById("selectArgument");
const selectArgumentScheme =
document.getElementById("selectArgumentScheme2");
const argumentForm = document.getElementById("counterArgumentScheme");

/*
 * Function which updates variable status of number of arguments to create
unique ID
 *
 */
db.collection("arguments").onSnapshot(function (querySnapshot) {
    numberOfArguments = querySnapshot.docs.length;
});

/*
 * Function which creates form and has event listener for submission of form
 * Also recognises if user is submitting counter-attacking argument
 *
 */
var createArgumentFromConsequencesForm = (function (elementToAppend, id,
buttonClass, buttonId, modalName, argumentStatus) {
    lib.createArgumentForm(elementToAppend, "If A is brought about good/bad
consequences will occur...", "afcPremise");
    lib.createArgumentForm(elementToAppend, "A should or should not be
brought about...", "afcConclusion");

    lib.appendArgumentButton(elementToAppend, buttonClass, buttonId);
    var argumentSubmissionButton = document.getElementById(buttonId);

    argumentSubmissionButton.addEventListener("click", function () {
        afcSubmissionToDatabaseForm(id, modalName);

        if (argumentStatus == "counterArgument") {
            lib.counterArgumentSubmissionToDatabase(numberOfArguments);
        }
```

```
    });

});

/*
 * Function which performs submission to database of user inputted data
 *
 */
var afcSubmissionToDatabaseForm = (function (id, modalName) {
    var form = document.getElementById(id);

    var afcPremise = document.querySelector("#afcPremise");
    var afcConclusion = document.querySelector("#afcConclusion");

    var variables = []
    variables.push(afcPremise.value, afcConclusion.value);
    test.fullVariableTesting(variables);

    var argumentFromUser = "Premise: " + afcPremise.value.toLowerCase() +
    "<br></br>Conclusion: " + afcConclusion.value.toLowerCase();

    // console.log("Argument = ", argumentFromUser);

    /* Submit fields to database */
    db.collection("arguments").add({
        name: "argument" + numberOfArguments,
        scheme: "Argument from Consequences",
        argumentDescription: argumentFromUser,
        afcPremise: afcPremise.value.toLowerCase(),
        conclusion: afcConclusion.value.toLowerCase()
    });

    /*----- Reset modal fields after argument submission ----*/
    var instance = M.Modal.getInstance(modalName);
    instance.close();

    selectTypeOfArgument.selectedIndex = "reset";
    selectTypeOfArgument.disabled = false;
    selectArgumentScheme.selectedIndex = "reset";
    selectArgumentScheme.disabled = false;
    argumentForm.className = "hide";


    form.reset();

    afcPremise.remove();
    afcConclusion.remove();

    var argumentButton = document.getElementById("counterArgumentButton");
    argumentButton.remove();
});

/* ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
```

```
-------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------
Critical question functionality ----------------------------------------
------------------------------
 */

/*
 * Function which dynamically builds critical questions
 *
 */
var setupAfcCriticalQuestions = (function (data) {
    var afcPremise;
    var conclusion;

    data.forEach(function (d) {
        afcPremise = d.afcPremise,
            conclusion = d.conclusion
    });

    createAndAppendAfcCriticalQuestions(afcPremise, conclusion);
});

var createAndAppendAfcCriticalQuestions = (function (afcPremise, conclusion)
{
    for (let i = 1; i < 5; i++) {
        var tempCriticalQuestion = afcQuestionsSwitch(i, afcPremise,
conclusion);
        lib.addAndAppendOption(tempCriticalQuestion, i);
    }
});

var afcQuestionsSwitch = (function (questionNumber, afcPremise, conclusion) {
    switch (questionNumber) {
        case 1:
            return "How strong is the probability or plausibility that these
cited consequences will (may, might, must) occur?";
            break;
        case 2:
            return "What is the evidence supports the claim that these
consequences will (may, might, must) occur?";
            break;
        case 3:
            return "Are the consequences of the opposite value that ought to
be taken into account?";
            break;
        case 4:
            return "Is \"" + conclusion + "\" really good or bad?";
            break;
        default:
            "Select a scheme to generate critical questions";
    }

});

```

```
export{
    createArgumentFromConsequencesForm,
    setupAfcCriticalQuestions
};
```

C.10 javascript/schemes/argumentFromCorrelationToCause.js

```
1  import * as lib from "../library.js";
2  import * as test from "../test.js";
3
4  /* ------------------------------------------------------------------------
   ------------------------------------------------------------------------
   ------------------------
5   * ------------------------------------------------------------------------
   ------------------------------------------------------------------------
   ------------------------
6   * ------------------------------------------------------------------------
   ------------------------------------------------------------------------
   ------------------------
7   * ------------------------------------------------------------------------
   ------------------------------------------------------------------------
   ------------------------
8   * ------------------------------------------------------------------------
   ------------------------------------------------------------------------
   ------------------------
9   * ------------------------------------------------------------------
   Argumentation functionality ----------------------------------------------
   -------------------------
10  */
11
12 var numberOfArguments = 0;
13
14 const selectTypeOfArgument = document.getElementById("selectArgument");
15 const selectArgumentScheme =
   document.getElementById("selectArgumentScheme2");
16 const argumentForm = document.getElementById("counterArgumentScheme");
17
18 /*
19  * Function which updates variable status of number of arguments to create
   unique ID
20  *
21  */
22 db.collection("arguments").onSnapshot(function (querySnapshot) {
23
24     numberOfArguments = querySnapshot.docs.length;
25 });
26
27 /*
28  * Function which creates form and has event listener for submission of form
29  * Also recognises if user is submitting counter-attacking argument
30  *
31  */
32 var createArgumentFromCorrelationToCauseForm = (function (elementToAppend,
   id, buttonClass, buttonId, modalName, argumentStatus) {
33
34     lib.createArgumentForm(elementToAppend, "There is a correlation between
   A...", "accVariableA");
35     lib.createArgumentForm(elementToAppend, "and B...", "accVariableB");
36     lib.createArgumentForm(elementToAppend, "Therefore A causes B",
   "accConclusion");
37
38     lib.appendArgumentButton(elementToAppend, buttonClass, buttonId);
39     var argumentSubmissionButton = document.getElementById(buttonId);
40
41     argumentSubmissionButton.addEventListener("click", function () {
42         accSubmissionToDatabaseForm(id, modalName);
43
```

```
        if (argumentStatus == "counterArgument") {
            lib.counterArgumentSubmissionToDatabase(numberOfArguments);
        }
    });

});

/*
 * Function which performs submission to database of user inputted data
 *
 */
var accSubmissionToDatabaseForm = (function (id, modalName) {
    var form = document.getElementById(id);

    var variableA = document.querySelector("#accVariableA");
    var variableB = document.querySelector("#accVariableB");
    var conclusion = document.querySelector("#accConclusion");

    var variables = [];
    variables.push(variableA.value, variableB.value, conclusion.value);
    test.fullVariableTesting(variables);

    var argumentFromUser = "Correlation Premise: there is a positive correlation between " + variableA.value.toLowerCase() + " and " + variableB.value.toLowerCase()
    + "<br></br> Conclusion: " + conclusion.value.toLowerCase();
    // console.log("Argument = ", argumentFromUser);

    db.collection("arguments").add({
        name: "argument" + numberOfArguments,
        scheme: "Argument from Correlation to Cause",
        argumentDescription: argumentFromUser,
        variableA: variableA.value.toLowerCase(),
        variableB: variableB.value.toLowerCase(),
        conclusion: conclusion.value.toLowerCase()
    });

    /*----- Reset modal fields after argument submission ----*/
    var instance = M.Modal.getInstance(modalName);
    instance.close();

    selectTypeOfArgument.selectedIndex = "reset";
    selectTypeOfArgument.disabled = false;
    selectArgumentScheme.selectedIndex = "reset";
    selectArgumentScheme.disabled = false;
    argumentForm.className = "hide";


    form.reset();

    variableA.remove();
    variableB.remove();
    conclusion.remove();

    var argumentButton = document.getElementById("counterArgumentButton");
    argumentButton.remove();
});


```

```
/* ------------------------------------------------------------------------------------------------------------------------
 * ------------------------------------------------------------------------------------------------------------------------
 * ------------------------------------------------------------------------------------------------------------------------
 * ------------------------------------------------------------------------------------------------------------------------
 * ------------------------------------------------------------------------------------------------------------------------
 * ------------------------------------------------------------------ Critical question functionality ------------------------------------------------------------------
 */

/*
 * Function which dynamically builds critical questions
 *
 */
var setupAccCriticalQuestions = (function (data) {
    var variableA;
    var variableB;
    var conclusion;

    data.forEach(function (d) {
        variableA = d.variableA;
        variableB = d.variableB;
        conclusion = d.conclusion;
    });

    createAndAppendAccCriticalQuestions(variableA, variableB, conclusion);
});

var createAndAppendAccCriticalQuestions = (function (variableA, variableB, conclusion) {
    for (let i = 1; i < 4; i++) {
        var tempCriticalQuestion = accQuestionsSwitch(i, variableA, variableB, conclusion);
        lib.addAndAppendOption(tempCriticalQuestion, i);
    }
});

var accQuestionsSwitch = (function (questionNumber, variableA, variableB, conclusion) {
    switch (questionNumber) {
        case 1:
            return "Is there really a correation between \"" + variableA + "\" and \"" + variableB + "\"?";
            break;
        case 2:
            return "Is there any reason for thinking the correlation is anything more than coincidence?";
            break;
        case 3:
```

```
            return "Could there be some third factor, C, that is causing both \"" + variableA + "\" and \"" + variableB + "\"?";
            break;
        default:
            "Select a scheme to generate critical questions";
    }
});

export {
    createArgumentFromCorrelationToCauseForm,
    setupAccCriticalQuestions
};
```

C.11 javascript/schemes/argumentFromPositionToKnow.js

```
1  import * as lib from "../library.js";
2  import * as test from "../test.js";
3
4  /* --------------------------------------------------------------------------------------------------------------------
5   * --------------------------------------------------------------------------------------------------------------------
6   * --------------------------------------------------------------------------------------------------------------------
7   * --------------------------------------------------------------------------------------------------------------------
8   * --------------------------------------------------------------------------------------------------------------------
9   * ------------------------------------------------------------------------ Argumentation functionality ------------------------------------------------------------------------
10  */
11
12 var numberOfArguments = 0;
13
14 const selectTypeOfArgument = document.getElementById("selectArgument");
15 const selectArgumentScheme = document.getElementById("selectArgumentScheme2");
16 const argumentForm = document.getElementById("counterArgumentScheme");
17
18
19 /*
20  * Function which updates variable status of number of arguments to create unique ID
21  *
22  */
23 db.collection("arguments").onSnapshot(function (querySnapshot) {
24     numberOfArguments = querySnapshot.docs.length;
25 });
26
27 /*
28  * Function which creates form and has event listener for submission of form
29  * Also recognises if user is submitting counter-attacking argument
30  *
31  */
32 var createArgumentFromPositionToKnowForm = (function (elementToAppend, id, buttonClass, buttonId, modalName, argumentStatus) {
33     lib.createArgumentForm(elementToAppend, "a is in the position to know...", "afpPositionToKnow");
34     lib.createArgumentForm(elementToAppend, "a asserts that A is true or false...", "afpAssertion");
35     lib.createArgumentForm(elementToAppend, "A may be plausibly taken to be true or false...", "afpConclusion");
36
37     lib.appendArgumentButton(elementToAppend, buttonClass, buttonId);
38     var argumentSubmissionButton = document.getElementById(buttonId);
39
40     argumentSubmissionButton.addEventListener("click", function () {
41         afpSubmissionToDatabaseForm(id, modalName);
42
```

```
        if (argumentStatus == "counterArgument") {
            lib.counterArgumentSubmissionToDatabase(numberOfArguments);
        }
    });

});

/*
 * Function which performs submission to database of user inputted data
 *
 */
var afpSubmissionToDatabaseForm = (function (id, modalName) {
    var form = document.getElementById(id);

    var positionToKnow = document.querySelector("#afpPositionToKnow");
    var assertionPremise = document.querySelector("#afpAssertion");
    var conclusion = document.querySelector("#afpConclusion");

    // var argumentFromUser = positionToKnow.value + " -> " +
assertionPremise.value + " -> " + conclusion.value;

    var variables = [];
    variables.push(positionToKnow.value, assertionPremise.value,
conclusion.value);
    test.fullVariableTesting(variables);

    var argumentFromUser = "Position to know premise: " +
positionToKnow.value.toLowerCase() +
    "<br></br>Assertion premise: " + assertionPremise.value.toLowerCase() +
    "<br></br>Conclusion: " + conclusion.value.toLowerCase();

    // console.log("Argument = " + argumentFromUser);
    // console.log("Number of arguments = " + numberOfArguments);

    /* Submit fields to database */
    db.collection("arguments").add({
        name: "argument" + numberOfArguments,
        scheme: "Argument from Position to Know",
        argumentDescription: argumentFromUser,
        positionToKnow: positionToKnow.value.toLowerCase(),
        assertionPremise: assertionPremise.value.toLowerCase(),
        conclusion: conclusion.value.toLowerCase()
    });

    /*----- Reset modal fields after argument submission ----*/
    var instance = M.Modal.getInstance(modalName);
    instance.close();

    selectTypeOfArgument.selectedIndex = "reset";
    selectTypeOfArgument.disabled = false;
    selectArgumentScheme.selectedIndex = "reset";
    selectArgumentScheme.disabled = false;
    argumentForm.className = "hide";

    form.reset();

    positionToKnow.remove();
    assertionPremise.remove();
    conclusion.remove();

```

```
100     var argumentButton = document.getElementById("counterArgumentButton");
101     argumentButton.remove();
102 });
103 
104 /* ------------------------------------------------------------------------
    ------------------------------------------------------------------------
    ------------------------
105  * ------------------------------------------------------------------------
    ------------------------------------------------------------------------
    ------------------------
106  * ------------------------------------------------------------------------
    ------------------------------------------------------------------------
    ------------------------
107  * ------------------------------------------------------------------------
    ------------------------------------------------------------------------
    ------------------------
108  * ------------------------------------------------------------------------
    ------------------------------------------------------------------------
    ------------------------
109  * ------------------------------------------------------------------
    Critical question functionality ----------------------------------------
    ----------------------------
110  */
111 
112 /*
113  * Function which dynamically builds critical questions
114  *
115  */
116 var setupAfpCriticalQuestions = (function (data) {
117     var positionToKnow;
118     var assertionPremise;
119     var conclusion;
120 
121     data.forEach(function (d) {
122         positionToKnow = d.positionToKnow;
123         assertionPremise = d.assertionPremise;
124         conclusion = d.conclusion;
125     });
126 
127     createAndAppendAfpCriticalQuestions(positionToKnow, assertionPremise,
    conclusion);
128 });
129 
130 var createAndAppendAfpCriticalQuestions = (function (positionToKnow,
    assertionPremise, conclusion) {
131     for (let i=1; i<4; i++){
132         var tempCriticalQuestion = afpQuestionSwitch(i, positionToKnow,
    assertionPremise, conclusion);
133         lib.addAndAppendOption(tempCriticalQuestion);
134     }
135 });
136 
137 /*
138  * Variable information for function
139  * positionToKnow = a
140  * conclusion = A
141  */
142 var afpQuestionSwitch = (function(questionNumber, positionToKnow,
    assertionPremise, conclusion){
143     switch (questionNumber) {
```

```
144         case 1:
145             return "Is the \"" + positionToKnow + "\" really in a position to know if the \"" + conclusion + "\" is true or false?";
146             break;
147         case 2:
148             return "Is \"" + positionToKnow + "\" an honest, trustworthy and reliable source?";
149             break;
150         case 3:
151             return "Did the \"" + positionToKnow + "\" really assert the \"" + conclusion + "\"?";
152             break;
153         default:
154             "Select a scheme to generate a critical question";
155     }
156 });
157 
158 export {
159     createArgumentFromPositionToKnowForm,
160     setupAfpCriticalQuestions
161 };
```

C.12 javascript/schemes/criticalActionScheme.js

```
import * as lib from "../library.js";
import * as test from "../test.js";

/* ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------
Argumentation functionality ----------------------------------------------
------------------------
 */

var numberOfArguments = 0;

const selectTypeOfArgument = document.getElementById("selectArgument");
const selectArgumentScheme =
document.getElementById("selectArgumentScheme2");
const argumentForm = document.getElementById("counterArgumentScheme");


/*
 * Function which updates variable status of number of arguments to create
unique ID
 *
 */
db.collection("arguments").onSnapshot(function (querySnapshot) {
    // console.log("Number of arguments = " + querySnapshot.docs.length);
    numberOfArguments = querySnapshot.docs.length;
});

/*
 * Function which creates form and has event listener for submission of form
 * Also recognises if user is submitting counter-attacking argument
 */
var createCasForm = (function (elementToAppend, id, buttonClass, buttonId,
modalName, argumentStatus) {

    lib.createArgumentForm(elementToAppend, "In the current circumstance...",
"casCircumstance");
    lib.createArgumentForm(elementToAppend, "We should perform the
action...", "casAction");
    lib.createArgumentForm(elementToAppend, "Which would result in new
circumstances...", "casNewCircumstance");
    lib.createArgumentForm(elementToAppend, "Which will realise goal...",
"casGoal");
    lib.createArgumentForm(elementToAppend, "Which will promote value...",
"casValue");

    lib.appendArgumentButton(elementToAppend, buttonClass, buttonId);
```

```
    var argumentSubmissionButton = document.getElementById(buttonId);

    argumentSubmissionButton.addEventListener("click", function () {
        casSubmissionToDatabaseFromForm(id, modalName);

        if (argumentStatus == "counterArgument") {
            lib.counterArgumentSubmissionToDatabase(numberOfArguments);
        }
    });

});

/*
 * Function which performs submission to database of user inputted data
 *
 */
var casSubmissionToDatabaseFromForm = (function (id, modalName) {

    var form = document.getElementById(id);
    var casCircumstance = document.querySelector("#casCircumstance");
    var casAction = document.querySelector("#casAction");
    var casNewCircumstance = document.querySelector("#casNewCircumstance");
    var casGoal = document.querySelector("#casGoal");
    var casValue = document.querySelector("#casValue");

    var variables = [];
    variables.push(casCircumstance.value, casAction.value,
casNewCircumstance.value, casGoal.value, casValue.value);
    test.fullVariableTesting(variables);

    var argumentFromUser = "In the current circumstance: " +
casCircumstance.value.toLowerCase() +
    "<br></br>We should perform action: " + casAction.value.toLowerCase() +
    "<br></br>Which will result in new circumstance: " +
casNewCircumstance.value.toLowerCase() +
    "<br></br>Which will realise the goal: " + casGoal.value.toLowerCase() +
    "<br></br>Which will promote the value: " + casValue.value.toLowerCase()
+
    "<br></br>";

    // console.log("Argument = " + argumentFromUser);

    /* Submit fields to database */
    db.collection("arguments").add({
        name: "argument" + numberOfArguments,
        scheme: "Critical Action Scheme",
        argumentDescription: argumentFromUser,
        currentCircumstance: casCircumstance.value.toLowerCase(),
        action: casAction.value.toLowerCase(),
        newCircumstance: casNewCircumstance.value.toLowerCase(),
        goal: casGoal.value.toLowerCase(),
        value: casValue.value.toLowerCase()
    });

    /*----- Reset modal fields after argument submission ----*/
    var instance = M.Modal.getInstance(modalName);
    instance.close();

    selectTypeOfArgument.selectedIndex = "reset";
    selectTypeOfArgument.disabled = false;
```

```
    selectArgumentScheme.selectedIndex = "reset";
    selectArgumentScheme.disabled = false;
    argumentForm.className = "hide";

    form.reset();
    casCircumstance.remove();
    casAction.remove();
    casNewCircumstance.remove();
    casGoal.remove();
    casValue.remove();

    var argumentButton = document.getElementById("counterArgumentButton");
    argumentButton.remove();
});



/* ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------------------
Critical question functionality ----------------------------------------
----------------------------
 */

/*
 * Function which dynamically builds critical questions
 *
 */
var setupCasCriticalQuestions = (function (data) {
    var currentCircumstance;
    var action;
    var newCircumstance;
    var goal;
    var value;

    data.forEach(function (d) {
        currentCircumstance = d.currentCircumstance;
        action = d.action;
        newCircumstance = d.newCircumstance;
        goal = d.goal;
        value = d.value;
    });

    createAndAppendCriticalQuestions(currentCircumstance, action, goal,
value, newCircumstance);
});

```

```
144
145 var createAndAppendCriticalQuestions = (function (currentCircumstance,
    action, goal, value, newCircumstances) {
146     for (let i = 1; i < 17; i++) {
147         var tempCriticalQuestion = casCriticalQuestionsSwitch(i,
    currentCircumstance, action, goal, value, newCircumstances);
148         lib.addAndAppendOption(tempCriticalQuestion, i);
149     }
150 });
151
152
153 var casCriticalQuestionsSwitch = (function (questionNumber,
    currentCircumstance, action, goal, value, newCircumstances) {
154
155     switch (questionNumber) {
156         case 1:
157             return "Are the believed \"" + currentCircumstance + "\" true?";
158             break;
159         case 2:
160             return "Assuming the \"" + currentCircumstance + "\", does the
    \"" + action + "\" have the stated consequences?";
161             break;
162         case 3:
163             return "Assuming the \"" + currentCircumstance + "\" and that the
    \"" + action + "\" has the stated \"" + newCircumstances + "\", will the \""
    + action + "\" bring about the desired \"" + goal + "\"?";
164             break;
165         case 4:
166             return "Does the \"" + goal + "\" realise the \"" + value + "\"
    stated?";
167             break;
168         case 5:
169             return "Are there alternative ways of realising the same \"" +
    newCircumstances + "\"?";
170             break;
171         case 6:
172             return "Are there alternative ways of realising the same \"" +
    goal + "\"?";
173             break;
174         case 7:
175             return "Are there alternative ways of promoting the same \"" +
    value + "\"?";
176             break;
177         case 8:
178             return "Does doing the \"" + action + "\" have a side effect
    which demotes the \"" + value + "\"?";
179             break;
180         case 9:
181             return "Does the \"" + action + "\" have a side effect which
    demotes some other value?";
182             break;
183         case 10:
184             return "Does doing the \"" + action + "\" promote some other
    value?";
185             break;
186         case 11:
187             return "Does doing the \"" + action + "\" preclude some other
    action which would promote some other value?";
188             break;
189         case 12:
```

```
            return "Are the \"" + currentCircumstance + "\" as described
possible?";
            break;
        case 13:
            return "Is the \"" + action + "\" possible?";
            break;
        case 14:
            return "Are the \"" + newCircumstances + "\" as described
possible?"
            break;
        case 15:
            return "Can the desired \"" + goal + "\" be realised?";
            break;
        case 16:
            return "Is the \"" + value + "\" indeed a legitimate value?";
            break;
        default:
            "Select a scheme to generate critical questions";
    }

});


export {
    createCasForm,
    setupCasCriticalQuestions
};
```

C.13 javascript/schemes/slipperySlopeArgument.js

```
import * as lib from "../library.js";
import * as test from "../test.js";

/* ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------------------
Argumentation functionality ---------------------------------------------
-----------------------------
 */

var numberOfArguments = 0;

const selectTypeOfArgument = document.getElementById("selectArgument");
const selectArgumentScheme =
document.getElementById("selectArgumentScheme2");
const argumentForm = document.getElementById("counterArgumentScheme");

/*
 * Function which updates variable status of number of arguments to create
unique ID
 *
 */
db.collection("arguments").onSnapshot(function (querySnapshot) {
    numberOfArguments = querySnapshot.docs.length;
});

/*
 * Function which creates form and has event listener for submission of form
 * Also recognises if user is submitting counter-attacking argument
 *
 */
var createSlipperySlopeArgumentForm = (function (elementToAppend, id,
buttonClass, buttonId, modalName, argumentStatus) {
    lib.createArgumentForm(elementToAppend, "A is up for consideration as a
proposal that seems like it should be brought about", "ssFirstStepPremise");
    lib.createArgumentForm(elementToAppend, "Bringing about A would lead to B
which would lead to C and so forth", "ssRecursivePremise");
    lib.createArgumentForm(elementToAppend, "This would eventually lead to a
horrible outcome", "ssBadOutcomePremise");
    lib.createArgumentForm(elementToAppend, "A should definitely not be
brought about", "ssConclusion");

    lib.appendArgumentButton(elementToAppend, buttonClass, buttonId);
    var argumentSubmissionButton = document.getElementById(buttonId);

    argumentSubmissionButton.addEventListener("click", function () {
        ssSubmissionToDatabaseForm(id, modalName);
```

```
42
43          if (argumentStatus == "counterArgument") {
44              lib.counterArgumentSubmissionToDatabase(numberOfArguments);
45          }
46      });
47
48  });
49
50  /*
51   * Function which performs submission to database of user inputted data
52   *
53   */
54  var ssSubmissionToDatabaseForm = (function (id, modalName) {
55      var form = document.getElementById(id);
56
57      var firstStepPremise = document.querySelector("#ssFirstStepPremise");
58      var recursivePremise = document.querySelector("#ssRecursivePremise");
59      var badOutcomePremise = document.querySelector("#ssBadOutcomePremise");
60      var conclusion = document.querySelector("#ssConclusion");
61
62      var variables = [];
63      variables.push(firstStepPremise.value, recursivePremise.value,
    badOutcomePremise.value, conclusion.value);
64      test.fullVariableTesting(variables);
65
66      var argumentFromUser = "First step premise: " +
    firstStepPremise.value.toLowerCase() +
67          "<br></br>Recursive premise: " + recursivePremise.value.toLowerCase()
    +
68          "<br></br>Bad outcome premise: " +
    badOutcomePremise.value.toLowerCase() +
69          "<br></br>Conclusion: " + conclusion.value.toLowerCase();
70
71      // console.log("Argument = ", argumentFromUser);
72
73      /* Submit fields to database */
74      db.collection("arguments").add({
75          name: "argument" + numberOfArguments,
76          scheme: "Slippery Slope Argument",
77          argumentDescription: argumentFromUser,
78          firstStepPremise: firstStepPremise.value.toLowerCase(),
79          recursivePremise: recursivePremise.value.toLowerCase(),
80          badOutcomePremise: badOutcomePremise.value.toLowerCase(),
81          conclusion: conclusion.value.toLowerCase()
82      });
83
84      /*----- Reset modal fields after argument submission ----*/
85      var instance = M.Modal.getInstance(modalName);
86      instance.close();
87
88      selectTypeOfArgument.selectedIndex = "reset";
89      selectTypeOfArgument.disabled = false;
90      selectArgumentScheme.selectedIndex = "reset";
91      selectArgumentScheme.disabled = false;
92      argumentForm.className = "hide";
93
94
95      form.reset();
96
97      firstStepPremise.remove();
```

```
    recursivePremise.remove();
    badOutcomePremise.remove();
    conclusion.remove();

    var argumentButton = document.getElementById("counterArgumentButton");
    argumentButton.remove();
});

/* ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------
Critical question functionality ----------------------------------------
----------------------------
 */

/*
 * Function which dynamically builds critical questions
 *
 */
var setupSsCriticalQuestions = (function (data) {
    var firstStepPremise;
    var recursivePremise;
    var badOutcomePremise;
    var conclusion;

    data.forEach(function (d) {
        firstStepPremise = d.firstStepPremise,
            recursivePremise = d.recursivePremise,
            badOutcomePremise = d.badOutcomePremise,
            conclusion = d.conclusion
    });

    createAndAppendSsCriticalQuestions(firstStepPremise, recursivePremise,
badOutcomePremise, conclusion);
});

var createAndAppendSsCriticalQuestions = (function (firstStepPremise,
recursivePremise, badOutcomePremise, conclusion) {
    for (let i = 1; i < 4; i++) {
        var tempCriticalQuestion = ssQuestionsSwitch(i, firstStepPremise,
recursivePremise, badOutcomePremise, conclusion);
        lib.addAndAppendOption(tempCriticalQuestion, i);
    }
});

var ssQuestionsSwitch = (function (questionNumber, firstStepPremise,
recursivePremise, badOutcomePremise, conclusion) {
```

```
    switch (questionNumber) {
        case 1:
            return "What interveing propositions in the sequence linking \""
+ recursivePremise + "\" are actually given?";
            break;
        case 2:
            return "What other steps are required to fill in the sequence to
make it plausible?";
            break;
        case 3:
            return "What are the weakest links in the sequence, the places
where key critical questions need to be asked?";
            break;
        default:
            "Select a scheme and generate critical questions";
    }
});

export {
    createSlipperySlopeArgumentForm,
    setupSsCriticalQuestions
}
```

C.14 javascript/argument.js

```
/*
 * Importing argument schemes from schemes folder
 *
 */
import * as cas from "./schemes/criticalActionScheme.js";
import * as apo from "./schemes/appealToPopularOpinion.js";
import * as afa from "./schemes/argumentFromAnalogy.js";
import * as acc from "./schemes/argumentFromCorrelationToCause.js";
import * as afc from "./schemes/argumentFromConsequences.js";
import * as ss from "./schemes/slipperySlopeArgument.js";
import * as afp from "./schemes/argumentFromPositionToKnow.js";
import * as eos from "./schemes/appealToExpertOpinion.js";

/*
 *------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
--------- Key global variables ------------------------------------------
------------------------
 */


/* Using materialise library to initialise modal */
const modal1 = document.getElementById("modal1");
M.Modal.init(modal1);

const selectTypeOfArgument = document.getElementById("selectArgument");

const selectArgumentScheme =
document.getElementById("selectArgumentScheme2");

const argumentForm = document.getElementById("counterArgumentScheme");

const counterArgumentInputFields =
document.getElementById("counterArgumentInputFields");

var argumentStatus = null;


/*
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
------------------------
```

```javascript
 * -------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
 * -------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
 * -------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
 * ------------------------------------------------------------------------------ Functions needed for initial arguments -----------------------------------------------------------------------------------
 */


/*
 * Event listener which deduces whether user wants to create a counter argument or initial argument
 */
selectTypeOfArgument.addEventListener('change', (event) => {

    // console.log("Value selected is = ", selectTypeOfArgument.value);

    if (selectTypeOfArgument.value == "initialArgument") {
        selectTypeOfArgument.disabled = true; /* To remove - search for 'disabled' */
        argumentStatus = "initialArgument";
        argumentForm.className = "show";
        // selectArgumentSchemeEventListener next function

    } else if (selectTypeOfArgument.value == "counterArgument") {
        selectTypeOfArgument.disabled = true; /* ADDED RECENTLY */
        argumentStatus = "counterArgument";
        /*
         * selectArgumentSchemeEventListener next function
         * Piping in fields for counterargument using div counterargumentInputFields on modal
         */
        createInputFieldsForCounterArgument(counterArgumentInputFields, "Counter argument target name", "counterArgumentTargetName");
        appendSeeCriticalQuestionsButton(counterArgumentInputFields, "btn waves-effect white-text", "counterArgumentTargetButton");
        appendSelectCriticalQuestions(counterArgumentInputFields);
        appendConflictingClaims(counterArgumentInputFields);

        argumentForm.className = "show";

        counterArgumentEventListener();
    }

});

/*
 * Event listener which listens to type of argument scheme to be selected e.g. critical action scheme
 */
selectArgumentScheme.addEventListener('change', (event) => {
    console.log("Value selected is = " + selectArgumentScheme.value)

    if (selectArgumentScheme.value == "casForm") {
```

```
        selectArgumentScheme.disabled = true;
        cas.createCasForm(argumentForm, "counterArgumentScheme", "btn waves white-text", "counterArgumentButton", modal1, argumentStatus);

    } else if (selectArgumentScheme.value == "appealToExpertOpinion") {
        // console.log("Appeal to expert opinion selected");
        selectArgumentScheme.disabled = true;
        eos.createEosForm(argumentForm, "counterArgumentScheme", "btn waves white-text", "counterArgumentButton", modal1, argumentStatus);

    } else if (selectArgumentScheme.value == "appealToPopularOpinionForm") {
        // console.log("Appeal to popular opinion selected");
        selectArgumentScheme.disabled = true;
        apo.createAppealToPopularOpinionForm(argumentForm, "counterArgumentScheme", "btn waves white-text", "counterArgumentButton", modal1, argumentStatus);

    } else if (selectArgumentScheme.value == "argumentFromAnalogy") {
        // console.log("Argument from analogy selected");
        selectArgumentScheme.disabled = true;
        afa.createArgumentFromAnalogyForm(argumentForm, "counterArgumentScheme", "btn waves white-text", "counterArgumentButton", modal1, argumentStatus);

    } else if (selectArgumentScheme.value == "argumentFromCorrelationToCause") {
        // console.log("Argument from correlation to cause selected");
        selectArgumentScheme.disabled = true;
        acc.createArgumentFromCorrelationToCauseForm(argumentForm, "counterArgumentScheme", "btn waves white-text", "counterArgumentButton", modal1, argumentStatus);

    } else if (selectArgumentScheme.value == "argumentFromConsequences") {
        // console.log("Argument from consequences selected");
        selectArgumentScheme.disabled = true;
        afc.createArgumentFromConsequencesForm(argumentForm, "counterArgumentScheme", "btn waves white-text", "counterArgumentButton", modal1, argumentStatus);

    } else if (selectArgumentScheme.value == "slipperySlopeArgument") {
        // console.log("Slippery Slope Argument selected");
        selectArgumentScheme.disabled = true;
        ss.createSlipperySlopeArgumentForm(argumentForm, "counterArgumentScheme", "btn waves white-text", "counterArgumentButton", modal1, argumentStatus);

    } else if (selectArgumentScheme.value == "argumentFromPositionToKnow") {
        // console.log("Argument From Position to Know Selected");
        selectArgumentScheme.disabled = true;
        afp.createArgumentFromPositionToKnowForm(argumentForm, "counterArgumentScheme", "btn waves white-text", "counterArgumentButton", modal1, argumentStatus);

    }


});
```

```
/* ------------------------------------------------------------------------------------------------------------------------
 * ------------------------------------------------------------------------------------------------------------------------
 * ------------------------------------------------------------------------------------------------------------------------
 * ------------------------------------------------------------------------------------------------------------------------
 * ------------------------------------------------------------------------------------------------------------------------
 * ------------------------------------------------------------------------------------------------------------------------
 * ------------------------------------------------------------------ Counter-argument functionality ------------------------------------------------------------------------------
 */

/*
 * Function to create input fields for counter-argument input on modal
 */
var createInputFieldsForCounterArgument = (function (elementToAppend, placeholder, id) {

    var inputField = document.createElement("input");

    inputField.setAttribute("type", "text");
    inputField.setAttribute("placeholder", placeholder);
    inputField.setAttribute("id", id);

    elementToAppend.append(inputField);
});

/*
 * Function which appends 'see critical questions' button to modal
 */
var appendSeeCriticalQuestionsButton = (function (elementToAppend, colour, id) {
    var div = document.createElement("div");
    div.setAttribute("class", "input-field");
    div.setAttribute("id", "buttonDiv");

    var button = document.createElement("btn");
    button.setAttribute("class", colour);
    button.textContent = "See critical questions";
    button.setAttribute("id", id);

    div.append(button);

    var div2 = document.createElement("div");
    div.append(div2);

    elementToAppend.append(div);
    // elementToAppend.append(button);
});

/*
 * Function which creates critical question opitions
```

```
 */
var appendSelectCriticalQuestions = (function (elementToAppend) {

    var label = document.createElement("label");
    label.textContent = "Select a critical question";
    label.setAttribute("id", "selectCriticalQuestionLabel");

    var select = document.createElement("select");
    select.setAttribute("class", "browser-default");
    select.setAttribute("id", "selectCriticalQuestion")

    var option = document.createElement("option");
    option.textContent = "Choose your option";

    select.append(option);

    elementToAppend.append(label);
    elementToAppend.append(select);
});

/*
 * Function which provides option for conflicting claims choice
 */
var appendConflictingClaims = (function (elementToAppend) {

    var label = document.createElement("label");
    label.textContent = "Are the claims conflicting";
    label.setAttribute("id", "conflictingClaimsLabel");

    var select = document.createElement("select");
    select.setAttribute("class", "browser-default");
    select.setAttribute("id", "selectConflictingClaims")

    var noOption = document.createElement("option");
    noOption.textContent = "No";

    var yesOption = document.createElement("option");
    yesOption.textContent = "Yes";

    select.append(noOption);
    select.append(yesOption);

    elementToAppend.append(label);
    elementToAppend.append(select);
});

/*
 * Function to create critical questions for user
 */
var counterArgumentEventListener = (function () {

    var counterArgumentTargetButton =
document.getElementById("counterArgumentTargetButton");
    var counterArgumentTargetName =
document.querySelector("#counterArgumentTargetName");

    // Will have to adapt to work for multiple different types of scheme
    counterArgumentTargetButton.addEventListener("click", function () {

        var counterArgumentTargetNameValue = counterArgumentTargetName.value;
```

```
        var args = db.collection("arguments");

        console.log("Counter argument target name = " +
counterArgumentTargetNameValue);

        // Generating critical questions - rename variables
        var query2 = args.where("name", "==",
counterArgumentTargetNameValue).get()
            .then(querySnapshot => {
                var scheme;

                query2 = querySnapshot.docs.map(doc => doc.data());
                // console.log(query2);
                // console.log("Empty array test on query = ",
emptyArrayTest(query2));

                query2.forEach(function (d) {
                    scheme = d.scheme;
                });

                // console.log("Scheme of selected = " + scheme);
                schemeSwitch(scheme, query2);
            });

    });

});

/*
 * Function which switches between critical questions for each argument
scheme
 */
var schemeSwitch = (function (scheme, data) {
    switch (scheme) {
        case "Critical Action Scheme":
            // console.log("cas activating");
            return cas.setupCasCriticalQuestions(data);
            break;
        case "Appeal to Expert Opinion":
            // console.log("eos activating");
            return eos.setupEosCriticalQuestions(data);
            break;
        case "Appeal to Popular Opinion":
            // console.log("apo activating");
            return apo.setupApoCriticalQuestions(data);
            break;
        case "Argument from Analogy":
            // console.log("afa activating");
            return afa.setupAfaCriticalQuestions(data);
            break;
        case "Argument from Correlation to Cause":
            // console.log("acc activating");
            return acc.setupAccCriticalQuestions(data);
            break;
        case "Argument from Consequences":
            // console.log("afc activating");
            return afc.setupAfcCriticalQuestions(data);
            break;
        case "Slippery Slope Argument":
            // console.log("ss activating");
```

```
            return ss.setupSsCriticalQuestions(data);
            break;
        case "Argument from Position to Know":
            // console.log("afp activating");
            return afp.setupAfpCriticalQuestions(data);
            break;
        default:
            console.log("Add switch functionality");
            return "Not a valid scheme";

    }
});

/*
 *-------------------------------------------------------------------------------------------------------------
 * -------------------------------------------------------------------------------------------------------------
 * -------------------------------------------------------------------------------------------------------------
 * -------------------------------------------------------------------------------------------------------------
 * -------------------------------------------------------------------------------------------------------------
 */
```

C.15 javascript/graph.js

```
/*
 *-------------------------------------------------------------------------
-------------------------------------------------------------------------
------------------------
 * -------------------------------------------------------------------------
-------------------------------------------------------------------------
------------------------
 * -------------------------------------------------------------------------
-------------------------------------------------------------------------
------------------------
 * -------------------------------------------------------------------------
-------------------------------------------------------------------------
------------------------
 * -------------------------------------------------------------------------
-------------------------------------------------------------------------
------------------------
 * ------------------------------------------------------------ Graph file
for plotting graph and applying labelling algorithm  ----------------------
------------------------
 */

/*
 * Data & firebase hook-up
 * Arguments denotes argument storage whilst links provides edges from
arguments
 */
var arguments = [];
var links = [];

/*
 * Function to connect and adapt on the basis of changes to database
 */
var databasePlusPlotGraph = (function (arguments, links) {

    var arguments = [];

    db.collection('arguments').onSnapshot(res => {

        res.docChanges().forEach(change => {

            const doc = {
                ...change.doc.data(),
                id: change.doc.id
            };

            switch (change.type) {
                case 'added':
                    arguments.push(doc);
                    break;
                case 'modified':
                    const index = arguments.findIndex(item => item.id ==
doc.id);
                    arguments[index] = doc;
                    break;
                case 'removed':
                    arguments = arguments.filter(item => item.id !== doc.id);
                    break;
                default:
                    break;
            }
```

```
        });

        update(arguments, links);

    });

    var links = [];

    db.collection('links').onSnapshot(res2 => {

        res2.docChanges().forEach(change => {

            const doc = {
                ...change.doc.data(),
                id: change.doc.id
            };

            switch (change.type) {
                case 'added':
                    links.push(doc);
                    break;
                case 'modified':
                    const index = data.findIndex(item => item.id == doc.id);
                    links[index] = doc;
                    break;
                case 'removed':
                    links = links.filter(item => item.id !== doc.id);
                    break;
                default:
                    break;
            }

        });


        // console.log(arguments);
        // console.log(links);

        update(arguments, links);
    });
});

var unlabelledArguments = [];
var inArguments = [];
var outArguments = [];
var undecidedNodes = [];

// console.log("Drawing graph");
databasePlusPlotGraph(arguments, links);
 /*
  * Logging database data to console
  */
console.log(arguments);
console.log(links);

/*
 * Event listener for complete labelling
 */
var labellingSwitch = document.getElementById("mySwitch");
let status = false;
```

```
107
108 labellingSwitch.addEventListener("change", function () {
109     var arguments = [];
110     var links = [];
111
112     // Status of labelling
113     status = labellingSwitch.checked;
114     databasePlusPlotGraph(arguments, links);
115 });
116
117 /*
118  * Event listener for grounded labelling
119  */
120 let grounded = false;
121 var groundedSwitch = document.getElementById("groundedSwitch");
122
123 groundedSwitch.addEventListener("change", function () {
124     var arguments = [];
125     var links = [];
126
127     grounded = groundedSwitch.checked;
128     databasePlusPlotGraph(arguments, links);
129 });
130
131 /*
132  * Event listener for preferred labelling
133  */
134 let preferred = false;
135 var preferredSwitch = document.getElementById("preferredSwitch");
136
137 preferredSwitch.addEventListener("change", function () {
138     var arguments = [];
139     var links = [];
140
141     preferred = preferredSwitch.checked;
142     databasePlusPlotGraph(arguments, links);
143 });
144
145
146 /*
147  * Draw graph using library and data with zoom functionality
148  */
149 const update = (arguments, links) => {
150     // Delete the old graph
151     d3.selectAll("svg > *").remove();
152
153     // Create the input graph
154     var graph = new dagreD3.graphlib.Graph().setGraph({});
155
156     // console.log("Status of complete labelling =", status);
157     // console.log("Status of grounded labelling =", grounded);
158     // console.log("Status of preferred labelling =", preferred);
159
160     /*
161      * If statement to decide what labelling algorithm to draw
162      */
163     if (status) {
164
165         unlabelledArguments = [];
166         inArguments = [];
```

```
        outArguments = [];
        undecidedNodes = [];
        completeLabelling(arguments, links, graph);

    } else if (grounded) {

        unlabelledArguments = [];
        inArguments = [];
        outArguments = [];
        undecidedNodes = [];
        groundedLabelling(arguments, links, graph);

    } else if (preferred) {

        unlabelledArguments = [];
        inArguments = [];
        outArguments = [];
        undecidedNodes = [];
        preferredLabelling(arguments, links, graph);

    } else {
        standardLabelling(arguments, graph);
        removeHeadersForInArguments();
    }

    graph.nodes().forEach(function (v) {
        var node = graph.node(v);
        node.rx = node.ry = 5;
    });

    links.forEach(function (l) {
        graph.setEdge(l.source, l.target, {
            curve: d3.curveBasis,
            minlen: 2
        });
    })

    // Create the renderer
    var render = new dagreD3.render()

    // Set up an SVG group so that we can translate the final graph.
    var svg = d3.select("svg"),
        inner = svg.append("g");

    var zoom = d3.zoom()
        .on("zoom", function () {
            inner.attr("transform", d3.event.transform);
        });

    svg.call(zoom);

    // Run the renderer. This is what draws the final graph.
    render(inner, graph);
};

/*
 * Standard labelling just presenting argumentative graph
 */
var standardLabelling = (function (arguments, graph) {

```

```
227     arguments.forEach(function (d) {
228         graph.setNode(d.name, {
229             labelType: "html",
230             label: "<b>" + d.name + "<br></br>" + "Scheme: " + d.scheme + "
    </b><br></br>" + d.argumentDescription,
231             class: "comp",
232         });
233     });
234 
235 });
236 
237 /*
238  * Complete labelling with nodes of argument labelled as IN / OUT contingent
    on criteria
239  */
240 var completeLabelling = (function (arguments, links, graph) {
241 
242     labellingAlgorithm(arguments, links);
243 
244     drawNodesWithArgumentLabellings(arguments, graph);
245 
246     addHeadersForInArguments();
247 });
248 
249 /*
250  * Grounded labelling with nodes of argument labelled as IN / OUT contingent
    on criteria
251  */
252 var groundedLabelling = (function (arguments, links, graph) {
253 
254     labellingAlgorithm(arguments, links);
255 
256     drawNodesWithGroundedLabellings(arguments, graph);
257 
258     addHeadersForInArguments();
259 });
260 
261 /*
262  * Preferred labelling with nodes of argument labelled as IN / OUT contingent
    on criteria
263  */
264 var preferredLabelling = (function (arguments, links, graph) {
265 
266     labellingAlgorithm(arguments, links);
267 
268     drawNodesWithPreferredLabellings(arguments, graph);
269 
270     addHeadersForInArguments();
271 });
272 
273 /*
274  *-------------------------------------------------------------------------
    -------------------------------------------------------------------------
    ------------------------
275  * -----------------------------------------------------------------------
    -------------------------------------------------------------------------
    ------------------------
276  * -----------------------------------------------------------------------
    -------------------------------------------------------------------------
    ------------------------
```

```
277  * ------------------------------------------------------------------------
     ------------------------------------------------------------------------
     ------------------------
278  * ------------------------------------------------------------------------
     ------------------------------------------------------------------------
     ------------------------
279  * ------------------------------------------------------------ Labelling
     algorithm to output graph nodes with IN / OUT  ----------------------------
     ------------------------
280  */
281
282 var labellingAlgorithm = (function (arguments, links) {
283     // console.log(links);
284     // console.log(arguments);
285     var targets = getTargets(links);
286     var sources = getSources(links);
287
288     unlabelledArguments = getArgumentNames(arguments);
289
290     var argumentLength = arguments.length;
291
292     // console.log("Argument length = ", argumentLength);
293     // console.log("Arguments being attacked = ", targets);
294     // console.log("Arguments attacking = ", sources);
295
296     // Test 1 - If an argument is not a target at all then it must be
     labelled as IN as no arguments are attacking it
297     unlabelledArguments = setExclusionBetweenArgumentsAndTargets(targets);
298
299     // console.log("In arguments after test 1 = ", inArguments);
300     // console.log("Unlabelled arguments after test 1 = ",
     unlabelledArguments);
301
302     for (let i = 0; i < argumentLength; i++) {
303     // Test 2 - If an argument is attacked by an IN argument it --> must be
     labelled as OUT
304         unlabelledArguments = argumentsAttackedByInArguments(links, sources);
305
306     // Test 3 - A4 on algorithm design tab --> an argument where ALL
     attacking it are OUT --> must be labelled as IN
307         unlabelledArguments = argumentsAttackedAllByOutArguments(links);
308
309     }
310
311     unlabelledArguments = labelRemainingNodesUndec(); // May need to perform
     tests again
312
313     // console.log("In arguments after test 4 = ", inArguments);
314     // console.log("Out arguments after test 4 = ", outArguments);
315     // console.log("Undec arguments after test 4 = ", undecidedNodes);
316     // console.log("Unlabelled arguments after test 4 = ",
     unlabelledArguments);
317
318 });
319
320 /*
321  * Function which returns all argument names
322  */
323 var getArgumentNames = (function (arguments) {
324     var argumentNames = [];
```

```
325
326     arguments.forEach(function (l) {
327         argumentNames.push(l.name);
328     });
329
330     return argumentNames;
331 });
332
333 /*
334  * Function which returns all arguments that are attacked
335  */
336 var getTargets = (function (links) {
337     var targets = [];
338
339     links.forEach(function (d) {
340         targets.push(d.target);
341     });
342
343     return targets;
344 });
345
346
347 /*
348  * Function which returns all arguments that are the source of attacks
349  */
350 var getSources = (function (links) {
351     var sources = [];
352
353     links.forEach(function (d) {
354         sources.push(d.source);
355     });
356
357     return sources;
358 });
359
360 /*
361  * Function which finds all arguments unattacked and labels them as IN
362  */
363 var setExclusionBetweenArgumentsAndTargets = (function (targets) {
364
365     inArguments = unlabelledArguments.filter(x => !targets.includes(x));
366     unlabelledArguments = unlabelledArguments.filter(inArguments =>
    targets.includes(inArguments));
367
368     return unlabelledArguments;
369 });
370
371 /*
372  * Function which finds all arguments attacked by IN arguments and labels
    them as OUT
373  */
374 var argumentsAttackedByInArguments = (function (links, sources) {
375     var inArgumentsAttacking = inArguments.filter(item =>
    sources.includes(item));
376     // console.log("In arguments attacking = ", inArgumentsAttacking);
377
378     links.forEach(function (d) {
379         if (inArgumentsAttacking.includes(d.source)) {
380             outArguments.push(d.target);
381         }
```

```
382     });
383
384     outArguments = removeDuplicates(outArguments);
385
386     unlabelledArguments = unlabelledArguments.filter(item =>
!outArguments.includes(item));
387     unlabelledArguments = unlabelledArguments.filter(item =>
!inArguments.includes(item));
388
389     return unlabelledArguments;
390 });
391
392 /*
393  * Function which finds all arguments attacked by only OUT arguments and
labels them as IN
394  */
395 var argumentsAttackedAllByOutArguments = (function (links) {
396     unlabelledArguments.forEach(function (currentNode) {
397
398         let argumentsAttackingCurrentNode = [];
399         argumentsAttackingCurrentNode = argumentsAttackingNode(currentNode,
links);
400
401         // console.log("Arguments attacking current node = ",
argumentsAttackingCurrentNode);
402
403         if (argumentsAttackingNodeAllOut(argumentsAttackingCurrentNode,
outArguments)) {
404             inArguments.push(currentNode);
405         }
406
407     });
408
409     unlabelledArguments = unlabelledArguments.filter(item =>
!outArguments.includes(item));
410     unlabelledArguments = unlabelledArguments.filter(item =>
!inArguments.includes(item));
411
412     return unlabelledArguments;
413 });
414
415 /* Function which identifies arguments attacking node */
416 var argumentsAttackingNode = (function (node, links) {
417     var argumentsAttackingNode = [];
418
419     links.forEach(function (d) {
420         if (d.target == node) {
421             argumentsAttackingNode.push(d.source);
422         }
423     });
424
425     return argumentsAttackingNode;
426 });
427
428 /*
429  * Function which checks if arguments attacking node are all OUT
430  */
431 var argumentsAttackingNodeAllOut = (function (argumentsAttackingNode,
outArguments) {
```

```
    return (argumentsAttackingNode.every(elem => outArguments.includes(elem)));
});

/*
 * Function which labels remaining nodes as UNDEC
 */
var labelRemainingNodesUndec = (function () {
    unlabelledArguments.forEach(function (currentNode) {
        undecidedNodes.push(currentNode);
    });

    unlabelledArguments = unlabelledArguments.filter(item => !outArguments.includes(item));
    unlabelledArguments = unlabelledArguments.filter(item => !inArguments.includes(item));
    unlabelledArguments = unlabelledArguments.filter(item => !undecidedNodes.includes(item));

    return unlabelledArguments;
});

/*
 *-------------------------------------------------------------------------------------------------------------------------
 * -------------------------------------------------------------------------------------------------------------------------
 * -------------------------------------------------------------------------------------------------------------------------
 * -------------------------------------------------------------------------------------------------------------------------
 * -------------------------------------------------------------------------------------------------------------------------
 * ------------------------------------ Accessory functions and code to draw svg nodes for user according to labelling   -------------------------
 */

/*
 * Function to remove duplicates
 */
var removeDuplicates = (function (chars) {

    let uniqueChars = chars.filter((c, index) => {
        return chars.indexOf(c) === index;
    });

    return uniqueChars;
});

/*
 * Function to draw nodes per complete argument labelling
 */
var drawNodesWithArgumentLabellings = (function (arguments, graph) {
    arguments.forEach(function (d) {
```

```
476         if (outArguments.includes(d.name)) {
477             setNodeWithOutLabelling(graph, d);
478         }
479         if (inArguments.includes(d.name)) {
480             setNodeWithInLabelling(graph, d);
481         }
482         if (undecidedNodes.includes(d.name)) {
483             setNodeWithUndecLabelling(graph, d);
484         }
485     });
486 });
487 
488 /*
489  * Function which applies drawing per grounded labellings
490  */
491 var drawNodesWithGroundedLabellings = (function (arguments, graph) {
492     arguments.forEach(function (d) {
493         if (outArguments.includes(d.name)) {
494             setNodeWithOutLabelling(graph, d);
495         }
496         if (inArguments.includes(d.name)) {
497             setNodeWithInLabelling(graph, d);
498         }
499         if (undecidedNodes.includes(d.name)) {
500             setNodeWithUndecGroundedLabelling(graph, d);
501         }
502     });
503 });
504 
505 /*
506  * Function which applies drawing per preferred labellings
507  */
508 var drawNodesWithPreferredLabellings = (function (arguments, graph) {
509     arguments.forEach(function (d) {
510         if (outArguments.includes(d.name)) {
511             setNodeWithOutLabelling(graph, d);
512         }
513         if (inArguments.includes(d.name)) {
514             setNodeWithInLabelling(graph, d);
515         }
516         if (undecidedNodes.includes(d.name)) {
517             setNodeWithUndecPreferredLabelling(graph, d);
518         }
519     });
520 });
521 
522 /*
523  * Function which applies node drawing per OUT labellings
524  */
525 var setNodeWithOutLabelling = (function (graph, d) {
526     graph.setNode(d.name, {
527         labelType: "html",
528         label: "<b>" + d.name + "</b>" + "<br></br><b>Label = OUT</b>" + "
    <br></br>" + d.argumentDescription,
529         class: "comp",
530         style: "fill: #ff6961",
531     });
532 });
533 
534 /*
```

```
 * Function which applies node drawing per IN labellings
 */
var setNodeWithInLabelling = (function (graph, d) {
    graph.setNode(d.name, {
        labelType: "html",
        label: "<b>" + d.name + "</b>" + "<br></br><b>Label = IN</b>" + "<br></br>" + d.argumentDescription,
        class: "comp",
        style: "fill: #77dd77",
    });
});

/*
 * Function which applies node drawing per UNDEC, IN/OUT labellings
 */
var setNodeWithUndecLabelling = (function (graph, d) {
    graph.setNode(d.name, {
        labelType: "html",
        label: "<b>" + d.name + "</b>" + "<br></br><b>Label = UNDEC, IN/OUT</b>" + "<br></br>" + d.argumentDescription,
        class: "comp",
        style: "fill: #fdfd96",
    });
});

/*
 * Function which applies node drawing per UNDEC labellings
 */
var setNodeWithUndecGroundedLabelling = (function (graph, d) {
    graph.setNode(d.name, {
        labelType: "html",
        label: "<b>" + d.name + "</b>" + "<br></br><b>Label = UNDEC</b>" + "<br></br>" + d.argumentDescription,
        class: "comp",
        style: "fill: #fdfd96",
    });
});

/*
 * Function which applies node drawing per IN/OUT labellings
 */
var setNodeWithUndecPreferredLabelling = (function (graph, d) {
    graph.setNode(d.name, {
        labelType: "html",
        label: "<b>" + d.name + "</b>" + "<br></br><b>Label = IN/OUT</b>" + "<br></br>" + d.argumentDescription,
        class: "comp",
        style: "fill: #77dd77",
    });
});

/*
 *-------------------------------------------------------------------------------------------------------------
 * -------------------------------------------------------------------------------------------------------------
 * -------------------------------------------------------------------------------------------------------------
```

```
-------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------------------------
------------------------------------------------------------------------
-------------------------
 * ------------------------------------------------------ Functions which
adapt headers below the svg argumentation graph   ------------------------
-------------------------
 */

/*
 * Function which adds headers below graph
 */
var addHeadersForInArguments = (function () {
    removeHeadersForInArguments();
    var switchDiv = document.getElementById("switchDiv");
    var groundedArgumentHeader = document.createElement("h6");
    var preferredArgumentHeader = document.createElement("h6");

    groundedArgumentHeader.innerText = "Grounded IN arguments: ";
    preferredArgumentHeader.innerText = "Preferred IN arguments: ";

    groundedArgumentHeader.setAttribute("id", "groundedArgumentHeader");
    preferredArgumentHeader.setAttribute("id", "preferredArgumentHeader");

    switchDiv.append(groundedArgumentHeader);
    switchDiv.append(preferredArgumentHeader);

    // console.log("In arguments to text...");
    inArgumentsToText();
});


/*
 * Function which removes headers below graph
 */
var removeHeadersForInArguments = (function () {

    var groundedArgumentHeader =
document.getElementById("groundedArgumentHeader");
    var preferredArgumentHeader =
document.getElementById("preferredArgumentHeader");

    if (groundedArgumentHeader) {
        groundedArgumentHeader.remove();
    }

    if (preferredArgumentHeader) {
        preferredArgumentHeader.remove();
    }
});

/*
 * Function which converts IN arguments to text
 */
var inArgumentsToText = (function () {
    inArguments.forEach(function (currentArgument) {
        appendInArgumentsToHeaderGrounded(currentArgument);
```

```
637         appendInArgumentsToHeaderPreferred(currentArgument);
638     });
639 
640     undecidedNodes.forEach(function (currentArgument) {
641         appendInArgumentsToHeaderPreferred(currentArgument);
642     });
643 });
644 
645 /*
646  * Function which appends grounded IN arguments
647  */
648 var appendInArgumentsToHeaderGrounded = (function (currentArgument) {
649     var groundedArgumentHeader =
document.getElementById("groundedArgumentHeader");
650     var tempHeader = document.createElement("p");
651     tempHeader.innerText = currentArgument;
652     groundedArgumentHeader.appendChild(tempHeader);
653 });
654 
655 /*
656  * Function which appends preferred IN arguments
657  */
658 var appendInArgumentsToHeaderPreferred = (function (currentArgument) {
659     var preferredArgumentHeader =
document.getElementById("preferredArgumentHeader");
660     var tempHeader = document.createElement("p");
661     tempHeader.innerText = currentArgument;
662     preferredArgumentHeader.appendChild(tempHeader);
663 });
664 
665 
666 
```

C.16 javascript/library.js

```
1
2  /*
3   *-------------------------------------------------------------------------------
    -------------------------------------------------------------------------------
    ------------------------
4   * -------------------------------------------------------------------------------
    -------------------------------------------------------------------------------
    ------------------------
5   * -------------------------------------------------------------------------------
    -------------------------------------------------------------------------------
    ------------------------
6   * -------------------------------------------------------------------------------
    -------------------------------------------------------------------------------
    ------------------------
7   * -------------------------------------------------------------------------------
    -------------------------------------------------------------------------------
    ------------------------
8   * -------------------------------------------------------------------------------
    --------- Library file for accesory functions  ----------------------------
    ------------------------
9   */
10
11
12 var createArgumentForm = (function (elementToAppend, placeholder, id) {
13
14     var inputField = document.createElement("input");
15
16     inputField.setAttribute("type", "text");
17     inputField.setAttribute("placeholder", placeholder);
18     inputField.setAttribute("id", id);
19
20     elementToAppend.append(inputField);
21 });
22
23 var appendArgumentButton = (function (elementToAppend, colour, id) {
24
25     var button = document.createElement("btn");
26
27     button.setAttribute("class", colour);
28     button.textContent = "Create argument";
29     button.setAttribute("id", id);
30
31     elementToAppend.append(button);
32 });
33
34 /*
35  * Function for creating links to add to collection within database
36  * Removing all necessary fields to clean up after submission
37  *
38  */
39 var counterArgumentSubmissionToDatabase = (function (numberOfArguments) {
40
41     var counterArgumentTargetButton =
    document.getElementById("counterArgumentTargetButton");
42     var counterArgumentTargetName =
    document.querySelector("#counterArgumentTargetName");
43     var buttonDiv = document.getElementById("buttonDiv");
44     var currentArgument = "argument" + numberOfArguments;
45
```

```
46      // console.log("Number of arguments at creation of links = ",
    numberOfArguments);
47
48      var conflictingClaimsLabel =
    document.getElementById("conflictingClaimsLabel");
49      var selectConflictingClaims =
    document.getElementById("selectConflictingClaims");
50
51      /* Creating links to add to collection within database */
52      // console.log("Counter argument target name = " +
    counterArgumentTargetName.value);
53      // console.log("Current counter-argument name = " + currentArgument);
54      createLinksForCounterArgument(currentArgument,
    counterArgumentTargetName.value);
55
56      // console.log("Conflicting claims selected value = ",
    selectConflictingClaims.value);
57
58      if (selectConflictingClaims.value == "Yes") {
59          // console.log("Drawing conflicting claims attack");
60          createLinksForCounterArgument(counterArgumentTargetName.value,
    currentArgument);
61      }
62
63      /* Resetting fields from counterargument modal */
64      var selectCriticalQuestion =
    document.getElementById("selectCriticalQuestion");
65      var selectCriticalQuestionLabel =
    document.getElementById("selectCriticalQuestionLabel");
66
67      // console.log("Critical question label = ",
    selectCriticalQuestion.value);
68      // selectCriticalQuestion.options.length = 0; --> can re-add if critical
    questions do not dissappear
69
70      counterArgumentTargetName.remove();
71      selectCriticalQuestion.remove();
72
73      conflictingClaimsLabel.remove();
74      selectConflictingClaims.remove();
75      buttonDiv.remove();
76
77      selectCriticalQuestionLabel.remove();
78      counterArgumentTargetButton.remove();
79  });
80
81  var createLinksForCounterArgument = (function (source, target) {
82      db.collection("links").add({
83          source: source,
84          target: target
85      });
86  });
87
88  var addAndAppendOption = (function (criticalQuestion, valueNumber) {
89      var criticalQuestionsForSelection =
    document.getElementById("selectCriticalQuestion");
90      var option = document.createElement("option");
91
92      option.value = criticalQuestion;
93      option.text = criticalQuestion;
```

```
    criticalQuestionsForSelection.add(option);
});

export {
    createArgumentForm,
    appendArgumentButton,
    counterArgumentSubmissionToDatabase,
    addAndAppendOption
};
```

C.17 javascript/test.js

```
1
2  /*
3   *-------------------------------------------------------------------------------
   ------------------------------------------------------------------------------
   ----------------------
4   * -------------------------------------------------------------------------------
   ------------------------------------------------------------------------------
   ----------------------
5   * -------------------------------------------------------------------------------
   ------------------------------------------------------------------------------
   ----------------------
6   * -------------------------------------------------------------------------------
   ------------------------------------------------------------------------------
   ----------------------
7   * -------------------------------------------------------------------------------
   ------------------------------------------------------------------------------
   ----------------------
8   * -------------------------------------------------------------------------------
   -------- Test file for usre input and unit testing  ------------------------
   ----------------------
9   */
10
11
12 var fullVariableTesting = (function (variables) {
13
14     // console.log("Test variables array = ", variables);
15
16     for (let i = 0; i < variables.length; i++) {
17         var currentVariable = variables[i];
18         if (emptyTest(currentVariable) || nullTest(currentVariable)) {
19             alert("Invalid argument variables input by user");
20             window.location.reload();
21         }
22     }
23
24 });
25
26 var emptyTest = (function (input) {
27     if (nullTest(input) == false) {
28         if (input.length > 0) {
29             return false;
30         }
31         return true;
32     }
33     return true;
34 });
35
36 var nullTest = (function (input) {
37     if (input == null) {
38         return true;
39     }
40     return false;
41 });
42
43 var emptyArrayTest = (function (input) {
44     if (input.length > 0) {
45         return false;
46     }
47     return true;
48 });
```

```
49
50
51 export {
52     fullVariableTesting
53 };
```