Heinrich Heine
University
Düsseldorf

Untangling Internet Debate - Decentralization and Reuse of Arguments for Online Discussion Software

Inaugural-Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von
Alexander Schneider
geboren in
Krasnyj Jar, Kasachstan

Düsseldorf, April 2020

aus dem Institut für Informatik
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Berichterstatter:

1. Prof. Dr. Martin Mauve

2. Prof. Dr. Stefan Conrad

Tag der mündlichen Prüfung: 05.06.2020

This dissertation is dedicated to my mother and my late father, who gave up their old lives so that their kids could achieve greater things. I will be forever thankful.

Abstract

During the last few years, dialog-based discussions have been researched and corresponding systems implemented. Although they improve upon some typical problems of online-discussions, e.g. bad overview, they still lack in other areas. This dissertation was written to further the development in one of those areas: the argument as a reusable resource. Nowadays, a lot of discussions happen online. It is possible to find a near infinite number of outlets discussing the same issues over and over again. As a participant in those discussions, it is easy to tire out. Repetition is often needed, because the participants of one discussion do not know the arguments and the back and forth that was conducted in another discussion. This is why it would be of huge value to be able to import arguments and existing discussions into other discussions, thus linking them on a meta-level. Just like hyperlinks bind together the World Wide Web (WWW).

Since this approach is novel, we first evaluated its general feasibility. For this, we built a Peer-to-Peer network in the PeerFactSIM.KOM simulator, in which the peers could set policies of what types of content to accept. This was necessary because in a realistic scenario not all discussions would allow import of all content. Seeing that the peers could communicate unobstructed, the results are promising and show that the approach is viable. Building on that, we postulate challenges that need to be solved to enable a distributed argumentation network. The main challenges are 1) How is it possible to update decentralized arguments? 2) How to handle arguments that carry an implicit context, which is needed to understand the argument? 3) What is a suitable design that enables real users to interact with the network? We propose solutions for all postulated challenges and use those to build a first prototype software, named *Extensible Discussion Entity Network (EDEN)*. *EDEN* is designed to be easily deployable by providers of discussions, e.g. forums or news-media. Its design allows the exchange of single well-defined components to suit the needs of the providers. This choice was deliberately made to facilitate wide-spread adoption. With the goal of making it as universally usable as possible we also present a range of tools that allow for the integration with *EDEN*. For example, the dialog-based discussion engine *D-BAS* works with *EDEN* by default. Other tools like *discuss* make it possible to upgrade every website to an *EDEN* provider. For evaluation purposes we present a study which we conducted with over 60 participants. The study is designed to show that a dialog-based toolchain, which includes *EDEN*, performs well in a realistic scenario. Furthermore, we show which parts of our dialog-based approach can be improved to further user adoption.

With this dissertation, we lay the groundwork for a new type of network: The Argumentation Network, which much like the WWW empowers people to link together arguments as a resource – effectively creating a meta-graph of argumentation that spans across online discussions all over the world.

Zusammenfassung

Während der letzten Jahre wurden dialog-basierte Diskussionen erforscht und entsprechende Systeme implementiert. Obwohl diese einige typische Probleme von Onlinediskussionen, etwa schlechte Übersichtlichkeit, verbessern, bleiben andere typische Probleme unangetastet. Diese Dissertation wurde erstellt, um die Weiterentwicklung vorrangig in einem dieser Gebiete voran zu treiben: Dem Argument als wiederverwertbare Ressource. Wir leben in einer Welt, in der unentwegt online diskutiert wird. Es ist ohne weiteres möglich, eine nahezu unendliche Menge an Webseiten zu finden, auf denen die gleichen Themen immer wieder diskutiert werden. Als Teilnehmende solcher Diskussionen, fällt es Menschen oft schwer, nicht müde zu werden. Es muss sich häufig wiederholt werden, da andere Teilnehmer einer Diskussion nicht die Argumente und den Verlauf der anderen Diskussion zum selben Thema kennen. Das ist der Grund, weshalb es von großer Wichtigkeit wäre, wenn bereits bestehende Argumente und Diskussionen in andere importiert werden könnten. Dies bedeutet auch, dass die Diskussionen dadurch auch auf einer Metaebene verknüpft werden – ganz so wie es Hyperlinks im World Wide Web (WWW) bereits tun.

Weil dieser Ansatz neu ist, begutachten wir als erstes dessen Machbarkeit. Dazu bauen wir ein Peer-to-Peer Netzwerk in dem Simulator PeerFactSIM.KOM. In diesem Netzwerk können die teilnehmenden Knoten selbst bestimmen, welche Arten von Inhalten sie annehmen. Dies zu untersuchen ist notwendig, weil in einem realistischen Szenario nicht alle Diskussionsanbieter sämtliche Arten von Argumenten importieren würden. Die Ergebnisse sind vielversprechend und zeigen, dass dieser Ansatz durchaus machbar ist. Anschließend bauen wir darauf auf und definieren Herausforderungen, die es zu lösen gilt, um ein verteiltes Argumentnetzwerk zu ermöglichen. Die sind hauptsächlich: 1) Wie kann man verteilte Argumente aktualisieren? 2) Wie geht man mit Argumenten um, denen ein impliziter Kontext anhaftet, der nötig ist um das Argument zu verstehen? 3) Was ist eine angemessene Architektur, die benutzerfreundlich genug ist um eine weite Verbreitung zu ermöglichen? Wir schlagen Lösungen für alle postulierten Herausforderungen vor und nutzen diese, um einen ersten Prototypen namens Extensible Discussion Entity Network (EDEN) zu bauen. Unser Prototyp ist in einer Weise designt, die es ermöglicht, von jedem Anbieter von Onlinediskussionen benutzt zu werden. EDENs Design erlaubt auch den Austausch einzelner gut definierter Komponenten um den Anforderungen der Anbieter gerecht zu werden. Dies war eine bewusste Entscheidung, die wir getroffen haben, um eine weitläufige Verbreitung zu ermöglichen. Damit EDEN möglichst ohne weitere Umstände einsatzfähig ist, präsentieren wir eine Auswahl an unterstützender Software, die entweder mit EDEN integrierbar ist, oder standardmäßig bereits integriert ist. Die dialogbasierte Diskussionssoftware D-BAS zum Beispiel ist bereits in EDEN integriert. Andere Software wie *discuss*, erlaubt es, beliebige Webseiten zu einem EDEN-Nutzer aufzuwerten. Zu Evaluationszwecken präsentieren wir auch eine Studie, welche wir mit über 60 Teilnehmern durchgeführt haben. Diese Studie wurde entworfen um zu zeigen, dass dialogbasierte Ansätze, darunter EDEN, in einem realistischen Szenario gute Ergebnisse liefern. Weiterführend zeigen wir anhand der Studie, welche Teile unserer dialogbasierten Ansätze verbessert werden können um von Nutzern besser angenommen zu werden.

Mit dieser Dissertation legen wir den Grundstein für eine neue Art Netzwerk: Das Argumentationsnetzwerk. Damit geben wir, ähnlich wie das WWW, Menschen die Möglichkeit Argumente miteinander zu verbinden. Letzendlich wird damit ein Metagraph von Argumenten geschaffen, welcher sich über Onlinediskussionen auf der gesamten Welt spannt.

Acknowledgments

I have many people to thank who made this dissertation possible. First and foremost, I want to thank Martin Mauve for giving me the opportunity to work and conduct research under his guidance. A special mention goes to my colleague and office-neighbor Christian Meter. We do not always have the same opinions, but our discussions usually yield interesting and productive results. My gratitude goes to all other colleagues that provided discussions, comments, jokes, tips, guidance and just general socializing, be it during a coffee break or at the famous #1120. Thank you: Philipp Hagemeister, Norbert Goebel, Raphael Bialon, Andre Ippisch, Tobias Schröder, Daniel Neugebauer, Jens Bendisposto, Björn Ebbinghaus, Kalman Graffi, Thomas Spitzlei, Sabine Freese, Markus Brenneis, Dennis Frieß and undoubtedly countless others that I forgot. And of course the biggest thanks of all goes to my parents and my wife, for always having my back and for the huge amount of support over the years.

Contents

Chapter 1

Introduction

Discussions are – and always were – an essential part of human communication. As conversations were used by our early ancestors to tell stories around the fire in order to exchange important lessons, discussions are used to exchange ideas and beliefs. More formalized types of discussions, i.e. debates, developed to discuss opposing views. Debates are used frequently in our society: Every democratic country uses debates in their form of parliament. Scientists debate theories. And the United States of America even elevated debate to a recurring spectacle by hosting the often heated presidential candidates debate.

Argumentation and debate take on a multitude of forms. In international negotiations, arguments are pervasive and seem to have an effect on the outcome of the negotiations under certain circumstances, as shown by Grobe [9]. Similarly, arguments play a big part in law in general [23] and in law-making in particular [12]. Besides all the other fields, argumentation, as the language of science, is also a core staple of scientific reasoning and learning [40]. Argumentation is also pervasive in advertisements [39], safety systems [16], healthcare [10], and clinical decision-making [28].

Hutchby [11] has, by analyzing British talk-in radio, shown that discourse and arguments can even give one participant power over another. Vaughn [**vaughn2008power**] discusses the importance of argumentation and the ability to recognize and formulate arguments in the process of critical thinking. And in many parts of the world, discussions are used to facilitate a dialog between local governments and their citizens. These forms of discussion are not only held face-to-face but increasingly online.

Computerized discussions were first conducted with the introduction of the USENET [1] in 1980. Its functions are similar to a precursory form of Internet forums. Threaded discussions can be held by posting or reading items from groups, which usually were aggregators for some field of interest. With the rise of personal Internet connections, forums – mostly in the form of bulletin boards [29] – became the focal point of computerized discussions. Slowly, one by one, most of the world could now hold discussions online and debate important and not so important topics – days were spent debating whether Star Wars or Star Trek was the better Sci-fi franchise. Those forums were a reiteration of the USENET newsgroups. A forum is typically segmented into thematic sub-forums, which in turn host so-called threads. A user can start a new thread. Alternatively, they can participate in an existing discussion by adding their post, which is just a string of text that is appended to the end of the current thread. This form of online-discussions has not changed since. More or less popular successors like Google+

or the groups feature of Facebook still function in basically the same way.

Even by Facebook's popularity alone, it is clear to see that a big chunk of the world is currently able to participate in some form of online discussion. This brings many opportunities and can be seen as a big equalizer, since everybody with Internet access is potentially able to spread ideas and express their standpoint in any number of discussions to any number of participants and readers. These acts of discussion and debate inherently are an important component for democratic and open societies.

Following the importance of online discussions, prompts one to ask whether the current threaded and list-based forum-style, is the best possible form thereof. The research presented here was largely influenced and prompted by this question. To dive in further, it is important to first understand the current situation regarding online discussions.

1.1 Motivation and Problem Statement

Currently, there are several go-to places on the Internet to discuss any number of topics. One would be the comments section of the digital newspaper of choice. This type is often sought to voice an opinion over political topics, which are discussed in the corresponding article. Arguments brought forth by the author are used and enhanced or refuted to position the commenting person. Often, those comment sections are not even threaded, but simple chronologically ordered lists. Then there are the comments on Facebook or other social networks. They are quite similar to the newspaper comments sections except that they typically offer the capability to answer directly to any commenting person and contain even less content. Furthermore, there are threaded forums that still survive despite the ubiquity of social networks. Remaining forums regularly enforce strict sets of rules and are small communities devoted to any enthusiast hobby or subject.

Problems that arise from those list-based discussions are largely the same for all described mediums.

Opaqueness / Scale The number of comments can rise quickly. Whether 20, 100 or 1000 – with a rising number of comments it gets increasingly harder to follow the discussion itself. Sub-discussions and strands of argumentation are nearly impossible to follow entirely.

Redundancy Arising in part from opaqueness, but also because there is no inherent mechanism to structure the content of arguments and comments, redundancy takes place. The bigger a discussion becomes, the more redundant standpoints are reiterated.

Dominance Fostered by the linear structure of the list-based form, people start reading corresponding discussions at the top. This in turn promotes the comments that were made first. They start to dominate the discussion, because nearly everybody reads them. The comments somewhere in the middle of a big discussion – e.g. in place 73 out of 139 – are seldom read and responded to.

Missing Context When the comment-lists do not allow threads, it is very hard to clearly and contextually respond to any comment. Even if threading is present, the context is still hard to judge. Most often, the threading is capped to one level, which creates a scenario where reactions to reactions are discussed on the same level as the original reactions, thus muddling the context of the discussion. On platforms like Reddit.com [14], an infinite number of levels is allowed. In this case the number of levels increases fast, adding to opaqueness and dominance of earlier comments.

Naturally, people experiment with other forms of structuring discussions. Another form that is often used is the Pro-Con list, which is basically two threads. One is filled with arguments for some cause and one thread with arguments against the same cause. Although this slightly helps with the opaqueness, the problems of dominance, redundancy and missing context are still prevalent. While there are a number of different systems experimenting with forms of online-discussions, none of those is used as regularly as list-based discussions or Pro-Con lists.

A potential solution to the aforementioned problems could be the dialog-based approach, developed by Krauthoff et al. [18]. In dialog-based argumentation, the user is at all times only confronted with arguments that react to statements the user was debating before. For example: If the user is debating the Brexit and specifically wrote an argument about the Brexit's impact on the Northern Irish border, then, in a dialog-based system, the user will be presented with arguments that talk specifically about the border situation and all implications thereof. The arguments the user is confronted with were entered by other participants of the discussion in an earlier session with the system. This way, the user is having a more "natural" time-shifted dialog instead of scanning over list-based arguments. Dialog-Based Argumentation System (D-BAS), the first system to make use of dialog-based argumentation, shows that it solves the challenges of missing context fully and the other challenges partly. Opaqueness, for example, is lessened since the user is guided through the parts of the discussions which are of interest to them. On the other hand, if the user is interested in getting an overview of the discussion, opaqueness would still be prevalent.

Dialog-based discussions in general and the D-BAS system in particular were the status quo on which this contribution was made. Although D-BAS improves upon list-based discussions, there are cases where D-BAS does offer no satisfactory solution. Continuing with the example of Brexit: Since the British referendum in June 2016, an uncountable number of discussions about the Brexit where conducted. Several problems still persist when the same topic is discussed throughout a lot of different places:

Redundancy Although redundancy in one discussion can be reduced, the redundancy between different discussions on the same topic is still a factor. When discussing the Brexit in the comments section of the BBC, the NY Times, and a Facebook group devoted to the citizens of London, one will encounter similar arguments and will be forced to repeat ones opinion several times to participate. Once for each digital outlet.

Filter Bubbles Since every discussion provider draws a certain crowd, filter bubbles are likely to emerge with time [7]. For example, in the London Facebook-group, one will encounter far more anti-Brexit arguments than in the comments section of “The Sun”. There is no easy option for the reader to publish their arguments on several outlets at once, which in turn makes it more likely to be published in places where the opinion is prevalent anyway. Thus, filter bubbles are strengthened.

Outdated Arguments Arguments can become outdated as circumstances change and new facts come to light. For example, an argument that the EU costs the UK a certain amount of money a month can be easily outdated as new statistics are created showing the opposite. But since arguments are usually simple text entries, there is only the option to either change the argument in place and lose the “historic” discussion, or to let the argument persist in its outdated state. If the argument has been copied in the meantime, there is no possibility to update the copied versions as well.

To tackle the challenges of redundancy, filter bubbles and outdated arguments, while uniting discussions on multiple digital outlets, an argument network is needed. In such a network, arguments are not simply text-based contributions, but valuable, reusable and versioned resource objects.

1.2 Contributions

Now that we identified the need for a network of rich argument objects, questions and challenges arise that need to be discussed. For the remainder of the section, we will articulate those questions and answer them accordingly with the insights gained while working on the concept of argument networks.

Decentralized Data Exchange The first question that might come to mind is whether arguments, intrepreted as data, need to adhere to certain criteria; i.e., can instances automatically exchange data while only handling data that adheres to certain specifications? An argument might be tolerable by the guidelines and standards of one host, but not by the standards of another. In short, while respecting the rules set by all participating hosts, would it be possible to have a functioning and automatic exchange of arguments? The answer is trivially “yes” when a central authority can monitor the needs of the participating entities and orchestrate the exchange. In the context of arguments and other potential sensitive data, a central authority is no feasible option. Censorship and manipulation by the central authority are made too easy. Thus, we must reformulate our question: Is it possible to exchange argument data, while adhering to the preferences of every instance, without a central authority? We discussed this question in the paper shown in Chapter 3. To this purpose, we build a Peer-to-Peer (P2P) network that contains instances which are able to express their preferences regarding content. We simulate a scenario where every instance is assigned random preferences based on a long-tailed distribution gathered from real-world data. During the simulation we measure the success rate of automatic data-exchange between the instances. We find that preference-based exchange is indeed possible and feasible in a P2P network without any authority.

Challenges in Distributing Arguments After the general feasibility has been shown, the development of an argument network can be tackled. Before designing the network, we must sketch out important challenges. Thus, questions that need asking to progress are: What challenges need to be solved for an argument network? And building on that: How can they be solved?

Both question are answered in the paper presented in Chapter 4. The main challenges that we identified are the following:

1. Context-dependence of arguments
2. Updating of decentralized arguments
3. User-friendliness
4. Suitable network design

Arguments are not free of the context they were written in. Using the Brexit example once more, in a discussion between British citizens, an argument against leaving the EU could be: "We lose money and benefits by leaving the EU". This would be attacked by the argument: "This is not true, since we are a net payer". In a scenario where another country, like Greece, is debating to leave the EU, the former argument about losing money and benefits could be imported, because it fits. Now if the counter is automatically imported too, it is nonsensical, since Greece is currently not a net payer of the EU. The problem here is that the arguments use the pronoun "we". The context of who the "we" is, is provided implicitly by the discussion. But since the context changes when the discussion changes, the arguments are not in all cases valid anymore. Because of this, it is important to incorporate mechanisms into the argument network which do not automatically import supports and attacks to other imported arguments without checking them for implicit context.

Since facts can change, and errors happen, there needs to be a mechanism to update arguments. Since we do not want to erase the older versions, which would delete "historical" arguments, an update mechanism for arguments needs to be baked into an argument network. In our paper, we propose a versioning scheme similar to software versioning to accomplish this goal. User-friendliness and a suitable network-design are "soft" goals since they do not concern argument networks per se but concern usable software. The paper proposes a federated network building on the findings from Chapter 3 and a user-focused interface, e.g. Meter's *discuss* [25]. Armed with the important challenges in mind, we continue with the development of a solution.

Architecture Development How does the environment look like for which we are building a distributed argument network? The answer can be summarized as: There exist heterogeneous providers of social media, forums, newspapers and similar content. Those usually have their own sets of guidelines for acceptable content and curated discussions. Thus, we call all instances that fit this description *argument aggregators*. Aggregators are the main nodes of the desired argumentation network. As we pinpoint the target audience, we need to ask: How does a suitable architecture and software catering to typical argument aggregators look like? We propose Extensible Discussion Entity Network (EDEN) as an answer, which is described in more detail in Chapter 5. EDEN is designed with modularity in mind, to facilitate its use by aggregators. Each of its four core modules that fulfill roles like exchanging arguments and communicating with a database is exchangeable and customizable.

As noted previously, the need to update arguments is elementary, which prompted us to focus the software-side architecture around the solution to this challenge. The data-structure used has many attributes like system and network wide IDs as well as special attributes like a predecessor pointer, which are almost solely used to enable versioning. Each object is either unique or represents a gradual update to another object. In the latter case, a predecessor is always set. This way, it is trivial to reconstruct older versions of arguments and discussions. Furthermore, this enables the network aggregators to use differing versions of arguments in accordance with their guidelines while preserving the ability of automatic import completely. EDEN in itself is a "backbone" to a larger construct which is akin to nerves in the human body. It provides communications between the varying parts and coordinates them. Now that the communication central is defined and functioning, we can start to look for how to construct the whole body.

Exemplary Ecosystem We introduced dialog-based argumentation in general and our vision of distributed argumentation in particular. Following, we can ask how an ecosystem that uses the proposed technology in a real-world scenario needs to look like. For this, we worked on a ready-to-go software bundle which incorporates the different contributions we made. Any argument aggregator can spin up one instance to instantly be connected to the argumentation network. Part of the bundle are the following components: EDEN, *discuss* and D-BAS. EDEN, as already introduced, provides the automatic exchange of arguments between the networked providers, thus building an overarching meta-graph of all related discussions. *Discuss* is used as a front-end for the everyday user. The included version is configured to use EDEN integration, which allows users to incorporate foreign arguments through an intuitive interface. D-BAS is used in the background to execute the dialog-based discussion logic. Furthermore, an *elasticsearch* instance is provided to enable proposition of semantically related arguments to the end-user.

Evaluating the Work The last question that needs to be asked is: Does the proposed ecosystem work as intended? To this end, we evaluate the performance of the developed approach in a lab experiment. Currently, comment sections are the dominant form of discussions on the internet, which is why we compare our systems to them. We gathered 63 volunteers and separated them in two groups. Both groups were presented the exact same three articles focusing on a vegetarian lifestyle and the environment. The difference between the groups was the used commenting style. One part of the participants was presented a *discuss* interface coupled with an EDEN server in the background. The control group got the typical comment section at the end of the article. We let the participants read and comment on the articles for 30 minutes. This was followed by a 10-minute questionnaire used to evaluate satisfaction with the software. Before the experiment, we formulated hypotheses, which aimed at the usability of the software and the subjective feeling of how informative the discussion was. The results showed that the test-persons rated the usability of the dialog-based system lower than that of the comment section approach. This was to be expected, since the comment sections are well known to users and the dialog-based approach is still in an early phase. We could also observe that using the dialog-based approach, the participants generated more than double the amount of arguments per user than with a comment section. Concluding, this shows us that the approach is viable to conduct objective discussions on the Internet, while still in need of user experience improvements to make it more universally usable for every-day people.

1.3 Outline

The remainder of the dissertation is structured as follows. In the next chapter, we present the related and preceding work. In Chapter 3 we discuss the contributions regarding compliance network research. This is followed by a chapter about the advantages of reusable statements in an argument network. Chapter 5 provides the context for EDEN before the complete ecosystem of dialog-based argumentation is discussed in Chapter 6. To evaluate the preceding work, a field-study with the software from the ecosystem is discussed in Chapter 7. The dissertation is concluded in Chapter 8 by an overview of future work and closing words.

Chapter 2

Related Work

The related work is presented in three parts. First, we are taking a look at related federated networks that inspired the aggregator structure that EDEN uses. Then, we present different forms of online argumentation and their relation to our chosen approach. Finally, we are reviewing the most closely related ideas: Publications that laid groundwork for decentralized argumentation networks.

2.1 Federated Networks

A big part of this dissertation is based on federated networks. Following are some systems, that are of similar spirit to our research.

Mastodon [33] A social network which aims at removing commercialization from the social feed. The ideas behind Mastodon are very similar to those used in our discussion networks. Mastodon relies on a federation of providers, which themselves govern over the acceptability of the content published on their platform. Every user is free to choose a server to participate. Despite this, users are able to see and comment on content generated on other servers. Thus, Mastodon is a close intellectual relative of EDEN regarding the federation principles in the social space.

Shibboleth [27] A single-sign-on provider, where especially the authentication part is handled in a federated way. When a user wants to access some resource that requires authentication, they can use a Shibboleth provider, which does so based on the relationship to and data from the user. For example, a student could get access to lecture materials based on their student status. A lecturer could get access to grades instead. Shibboleth is especially relevant for this work, since future development of EDEN may contain decentralized user data storage.

Jabber [34] Similar to Mastodon, Jabber fulfills a social need. It is based on the XMPP protocol and its addressing of users is federated. A user can register with any Jabber service to receive a contact address. Given that they can be contacted, irrespective of the server anybody is registered with. The routing of the messages is handled by the services that federate with each other. Like Mastodon, Jabber is close to the principles of EDEN in regard to the federation ideas.

2.2 Forms of Online Argumentation

This section focuses on relevant forms of online argumentation. We describe differences and similarities of our work with the discussed approaches.

Disqus [5] Disqus is a typical implementation of a list-based discussion system. Users' comments and arguments are presented in chronological fashion. Directly addressing other users is possible for comments at the top-level of the discussion. Disqus enables different types of content-providers, like bloggers or journalists, to embed the system directly into their website. The principle is similar to *discuss* [25], although *discuss* additionally allows addressing specific sentences instead of commenting on the article as a whole. One of disqus' features is presenting relevant arguments from other discussions. In contrast to EDEN, disqus only presents similar arguments from the same provider instead of several. Furthermore, there is no easy way to logically reuse an argument or a part of the discussion.

Reddit [14] The website where most discussions are being held at any time is currently reddit. In 2013 already 6% of American adults, which used the Internet, also used reddit [6]. The discussion form is also a simple chronological list with the twist, that it allows for unbounded threading depth. This helps immensely with the problem of specificity. It is easy to see which specific statement a comment or argument is targeting. At the same time this is one of reddit's greatest weaknesses, because in a deeply threaded discussion, it is incredibly hard to visually follow it to its full extent. This opaqueness in turn enables repetition during discussions which we try to reduce to a minimum with our systems.

Kialo [13] A website, specially made for structured discussions of popular topics. It is similar to the tools presented in this dissertation in that its goal is making structured discussions available to layman users. The difference lies in the form of discussion. While we utilize dialog-based discussions, kialo is based on Pro-Con lists.

D-BAS [18] Most of the work in this dissertation is based on and extends the first implementation of dialog-based discussions: D-BAS. D-BAS works as a website, which presents a (layman-)user with the possibility of starting or joining discussions. They are confronted with arguments from other users and can answer while an argument graph is created and curated in the background. Users' comments can be entered through an interface, that nudges the user towards structuring their arguments into a premise and a conclusion. This simplifies the discussion process and makes the creation of a discussion graph in the background easy.

Deliberatorium [17] The MIT Deliberatiorium, akin to our presented ideas, targets everyday users. The general idea is to collaboratively create an argumentation map, which helps the users deliberate some issue. Whereas the Deliberatorium uses collaborative and moderated map-creation, we follow the same goals by simulating an argumentative dialog.

Araucaria [32] This system was developed to support the mapping of discussions and argumentation. It supports many modes, one of which is a dialogic argumentation. In contrast to this work, Araucaria is not a web app. It is also not aimed towards layman users, but rather at a more expert audience.

ConsiderIt [20] This web-based tool, enables normal users to submit Pro-Con arguments regarding a topic. In general the concept is not very different to e.g. kialo. One factor, where ConsiderIt differs, is the visual representation of the users arguments. With the accompanying tools, users can express their opinion regarding a topic as a point on a spectrum.

2.3 Decentralized Argumentation

The main goal of this dissertation was to create a system and an environment, in which arguments and discussions are shared resources. There are other approaches and systems which were built with the same goal in mind, setting the stage for our work.

To set the context: At the beginning of the 2000s and before, many theoretical frameworks for the representation of argumentation had been developed. Most of them were not compatible with each other, which made it hard to establish something like web native arguments. To solve this problem and establish a universal and versatile framework, the Argument Interchange Format (AIF) has been published by Chesnevar et al. [4]. Since the first draft was only an abstract common ground specification, Rahwan and Reed [30] published some examples showcasing the abilities of AIF.

After the establishment of an interchange format, creation of software was the next logical step. Rahwan et al. [31] published a concept for a World Wide Argument Web and a first example application called ArgDF, which allowed for creation and interlinking of arguments on the network. Bex et al. [2] followed up on this with the presentation of a first full prototype. The ecosystem around the argument web enabled applications like ArguBlogging [3], which gave users the power to convert arguments on the web into semantically enriched argument micro-blogs, or Arvina [21] which gave users an interface into the argument web.

The central application for the argument web, which shares the most roots with this dissertation, was the AIFdb [22]. AIFdb was the first application which used an interface for storage and retrieval of arguments. It therefore was the first application which classified arguments on the internet as a commodity. Our work, especially EDEN, builds on the same ideas, improving them for example by decentralizing the argument database. Although EDEN by default does not use AIF, it is trivial to adapt one of its module to enable this capability.

Chapter 3

Compliance Management for P2P Systems

This chapter summarizes the contributions of the paper [35]:

Alexander Schneider, Martin Mauve:
Compliance Management for P2P Systems
In Proceedings of 23rd Asia-Pacific Conference on Communications (APCC 2017),
Perth, Australia, December 2017.
Acceptance ratio: 70%.[1]

P2P networks have been widely used for the exchange of data through multitudes of different file-sharing networks. More recently, P2P networks have seen a rise in popularity through the introduction of the WebRTC [15] standard and its inclusion in most popular web-browsers as well as through the adoption of crypto-currencies which often rely heavily on users participating in a P2P network.
Nonetheless, most people associate P2P networks with illegal or at least ethically problematic actions like sharing pirated software, movies and series. Additionally, the health of most P2P networks relies on users actively participating in the networks purpose (i.e. sharing files) and thus often penalizes users for not participating. To help users overcome the problem of handling unwanted data we propose the development of a compliance management system for P2P networks. Users in such a network are enabled in stating explicit preferences about which data they are willing to store and forward. The network adapts to users wishes and constructs its routing around their preferences. The goal of this paper is to introduce the idea of compliance management in P2P networks, create a prototype and show its feasibility through simulations.

3.1 Paper Summary

This paper explores the concept of a P2P network with build-in compliance management. Three key contributions are:

1. The introduction of the concept of compliance-based routing. We motivate the need and define challenges that need to be overcome to get a preliminary understanding of this new mechanism.

2. The prototype architecture for a compliance routing enabled P2P network based on Kademlia [24].

3. A simulation we conducted which shows that the devised prototype is feasible and by extension compliance management in P2P networks.

The paper introduces and motivates the concept of compliance-based routing in its first chapter. It is stated that P2P networks have a bad reputation, because they are often used to exchange data that may be illegal at the users residence or ethically concerning to some users. At the same time it is critically important that most users of a network participate in transporting the data to keep it healthy and functioning. This fundamental conflict deters a number of users from participating. Related work is presented in two categories. One is the class of P2P networks enabling the storage and exchange of data inside the network. The second part of related work discusses adding a categorization to the exchanged data, i.e. through tags. It is also highlighted that none of the categorization systems allow to cryptographically, or otherwise, secure the links between a category and some data.

Following that, we present Comademlia, our prototype for a compliance management enabled P2P network. It uses the Kademlia overlay as a base and extends its routing mechanisms. In Comademlia, data is enriched with tags that describe the underlying data. The user then can either use a blacklist approach and define which tags they are not willing to handle or a whitelist approach where they explicitly define all tags they will accept. This preference is broadcast to all neighboring nodes in the network. The nodes then cultivate a routing table for every tag they are willing to handle. Filling and maintaining of the routing tables works analogous to Kademlia, with the exception that Kademlia only maintains a single routing table.

The feasibility of the prototype and by extension the concept of a compliance management network is shown by a simulation. We used the PeerfactSIM.KOM [8] simulator and modified its Kademlia layer to reflect the changes described by our Comademlia prototype. To gain tag distributions which model real world data as closely as possible, we queried the top 50 tags on Q&A site stackoverflow.com. We used this data to model the distribution of tags on data inside the simulation. The simulated network consisted of 256 nodes on every run-through and every node preferred up to 10 tags that it accepted. Every of the ten simulations ran with a differing seed and for 24 hours of simulated time. The results of the simulations show that the accessibility of the data was near 100% despite the content restriction. Furthermore, the network complexity was shown to scale with the number of tags a node accepts on average. A negative result was the high number of messages inside the network, that were needed to keep it functional. This was probably due to the naive implementation of Comademlia, which was not optimized for performance.

Thus, we conclude that compliance based P2P networks are a feasible concept as showed by the conducted simulations.

3.2 Contribution

The first key contribution of this paper is the introduction of compliance based routing as a concept. Prior to our paper P2P overlays were optimizing routing only toward technical metrics like speed, hop-count or transfer-rate. Compliance based routing on the other hand optimizes for the user preferred content categories. This is aimed at improving participation of everyday users in P2P networks.

The second contribution is the devising of an overlay, which is based on Kademlia and implements the previously defined compliance-based routing. This overlay, named Comademlia is tested in a simulator and shows that compliance based routing is a feasible concept for P2P networks and should be further looked into.

3.3 Personal Contribution

Alexander Schneider, the author of this thesis was the main contributor of the paper. He developed the notion of a network that routes compliant to the wishes of the users regarding the routed data. Furthermore, he developed the corresponding prototype software and tested it in the PeerfactSIM.KOM Simulator. The analysis of the simulation results and the writing of the paper was also performed by him.

Martin Mauve provided feedback regarding editorial choices for the paper and discussed the ideas with the main author.

3.4 Importance and Impact on the Thesis

The paper presented in this chapter lays the groundwork for the inception of EDEN which is itself presented in Chapter 5. We developed the notion of data-routing, which is based on the compliance with a users wishes. This is a central concept that leads to the notion of distributing arguments while respecting the community standards of content-aggregators as practiced in EDEN. Without the insights gained from compliance management, we would not have been able to propose blacklists for EDEN without disturbing the network. Furthermore, we not only planted the seeds for the later notion of policy based data exchange in distributed systems, but also proved the concept to be a functioning one with regard to realistic data. The following chapters thus show that the exchange of arguments through federated networks is strongly related to compliance based routing in P2P networks.

Compliance Management for P2P Systems

Alexander Schneider
Institute for Computer Science
Univeristy Düsseldorf
Düsseldorf, Germany
Email: aschneider@cs.uni-duesseldorf.de

Martin Mauve
Institute for Computer Science
Univeristy Düsseldorf
Düsseldorf, Germany
Email: mauve@cs.uni-duesseldorf.de

***Abstract*—Compliance management for peer-to-peer networks describes a process ensuring that content inside the network is distributed and stored in a way that does not violate user defined preferences. Several use cases, ranging from filesharing networks to distributed computing and content delivery networks, can be enhanced with compliance management. To our knowledge there are no existing peer-to-peer architectures which allow for compliance management. In this paper we propose an architecture, which utilizes policy-based routing and storage as well as a categorization of content in order to provide compliance management. We implement a prototype and evaluate it through simulations to show that compliance management in peer-to-peer networks is actually feasible.**

I. Introduction

Peer-to-Peer (P2P) networks enable users to exchange a wide variety of information. Participation in those networks typically requires that each user is willing to forward and store arbitrary data. This may cause legal problems or, in a somewhat less severe case, raise ethical concerns. To solve this problem we introduce the idea of peer-to-peer networks with integrated compliance management. In those networks users are only involved in managing data that they explicitly agree to handle.

One example application is the distribution and storage of arbitrary data, such as music or video files, as it is common in many current peer-to-peer networks. With compliance management each participant would specify the content she is willing to forward and store. Hence she could participate and support the network without risk of breaking the law or handling unwanted content. Another example is a social network, managed and maintained by a single company which uses a peer-to-peer network for distributed storage. The nodes of the peer-to-peer network could be located in distinct countries each with their own jurisdiction and customs. The company then needs to make sure that each node in the peer-to-peer network stores and manages only data that is legal and acceptable in the country the peer is located in. This allows the social network to maintain content that is acceptable in any country it operates in, instead of restricting content to what all countries deem as acceptable.

In order to specify which content is acceptable for a given node, we propose that each chunk of data handled by the peer-to-peer network is assigned one or more tags describing its content. We acknowledge that assigning these tags in a reliable and trustworthy way is a significant challenge that we do not address at length in this paper. We will, however, provide reasoning why we believe this to be a solvable problem. Given a chunk of data that is described by a list of tags, we seek to answer the following question: is it possible to store and forward it in such a way, that none of the individual preferences of the users are violated? In this context the preferences of a user are given as a list of tags, that she is willing to forward and store, while the network organizes itself in a way that incorporates the participants preferences. In this paper we answer this question by adapting Kademlia to use compliance management. We show that compliance management in peer-to-peer networks is, in fact, feasible and we provide insights on the impact that the preferences of the participants have on the performance of the network.

The remainder of this paper is structured as follows. Section II briefly reviews related work. Following this, we introduce a modified version of Kademlia - Comademlia - that is able to provide compliance management in section III. In section IV we evaluate the performance of Comademlia for several parameters and enhancements. Lastly, section V concludes the paper and gives an outlook on future work.

II. Related Work

There are many peer-to-peer systems that can serve as distributed storage. Approaches such as Kadmelia [5], Chord [11] or Pastry [7] serve this purpose very well. However none of them allow for compliance management by the participating nodes. There are also filesystem-like destributed storage solutions like Oceanstore [4] and Ivy [6]. They, however, also do not provide compliance management.

Da Silva et al. [1] published work, that outlines policy based access in P2P grids. In contrast to our architecture it only regulates the access to data based on policies, but does not use policies to distribute and store the data. To our best knowledge there are currently no systems that use content policies to determine storage and forwarding rules for individual nodes inside a peer-to-peer network.

There exists work regarding peer-to-peer networks, which utilize content tags, e.g. [2], [3]. The existing systems however define the possibility to share data annotated with tags and the ability to calculate and maintain feature-vectors in the tagging environment, while missing out on a cryptographic link between content and tag. Other work by Smetters and Jacobson [9] introduces the idea to cryptographically link arbitrary names to content, which is discussed later in the paper.

III. Comademlia

Compliance management requires that a node can somehow judge the content contained in a chunk of data. We believe that this can be done, e.g. by annotating each chunk of data with tags. The process of assigning those tags is not part of the work we describe here. However, we briefly reason, why this is a solvable problem. The focus of our work, then, is the actual storage and routing, under the constraint that each node only participates in tasks that do not violate its local policy.

A. Tagging

We assume that all chunks of content are associated with tags, that characterize the content. Tags either describe the content, e.g. “violence” or “explicit speech”, or they provide meta information such as “legal for all audiences in Germany”. They are assigned by trusted parties, like the publisher of the content, or by means of collective decisions. The latter is very similar to what is regularly done in order to realize quality control at popular web-sites. Each node in the peer-to-peer network specifies its policy by maintaining a list of tags and announcing it to its neighbors. A node will not participate in routing and storing content, with tags that are not contained in its policy. Of course the link between a tag and some data has to be trustworthy, which can be achieved by e.g. cryptographic signatures. For example, it was showed in [9] that it is possible to establish cryptographic links between names and content.

A real-world implementation needs mechanisms, to report and remove tags, since data can be falsely tagged, either by accident or by malice. In our prototype we did not include such a mechanism, but would like to sketch a possible solution. The general idea is to allow reporting of assumed false tags only if the reporting node is trustworthy. To find out whether a node is trustworthy, we could asses how cooperative the node is with the desired network operations. The more a node forwards and stores content, the more it is contributing to the health of the network, so either the nodes is an honest participant or it contributes for the purpose of being able to issue false reports. Ultimately if a node wants to report tags as incorrect, it has to “expend” some of the accumulated cooperation. The idea is to use cooperative actions as a resource in a proof-of-work-like system. We assume hereby that the threshold for a report can be adjusted to a level, where nodes with malicious and false reports add a significant contribution to the network, that balances out their wrongdoing. Individual nodes can then configure the amount of valid reports needed until they stop trusting the tags of a certain content publisher. The reports can be organized in different systems, e.g. in an overlay or using a blockchain. Developing such a system is out of scope for this paper, but constitutes our main focus for future work.

A good and efficient system for the assignment of tags - in particular in form of collective decisions - is certainly an interesting research challenge. However, given that content classification is regularly done in other contexts both in a centralized fashion and as collective decisions, leads us to the assumption that developing such a system is generally feasible. In the remainder of the paper we therefore focus on the networking aspects of a peer-to-peer system that provides compliance with the preferences of the individual users.

B. Distribution and Storage

Our goal is to design a peer-to-peer-based content storage network prototype that provides permanence, high availability of data and, most importantly, enforces compliance with the policies of each individual participant. As a starting point we used Kademlia and modified it to include compliance management.

1) Kademlia: In a Kademlia [5] network nodes are assigned a random identifier inside an ID space. Data is also assigned a value from the same ID space, by hashing the data. Inside this ID space XOR is used as a metric to determine the distance between two IDs, which is important for routing decisions.

Every Kademlia node uses a tiered routing table, with ”buckets”. A bucket contains a limited number of nodes which share a certain ID prefix. All buckets together cover the entire ID space without overlap or gaps. The buckets are organized in a way, that facilitates more complete knowledge of nodes in the immediate vicinity. Vicinity in this case is defined through a low XOR-distance.

Data is always redundantly stored at a set of nodes whose IDs are closest to the ID of the data. Thus, storing and retrieving data is about finding one or multiple nodes nearest to a certain ID. To find nodes, Kademlia uses an iterative lookup process. A node starts the lookup by querying the nearest known nodes to the desired ID. Those nodes then return their nearest known nodes, in respect to the target ID, to the requester. The process is repeated until no new nodes are returned and a sufficient subset of nodes has been queried or the desired target, or data, has been found. Since every node has extensive knowledge of its surroundings the process always converges with time.

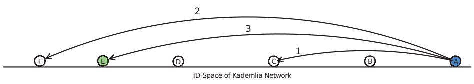

Fig. 1. The Kademlia lookup process illustrated.

An example for a lookup can be found in figure 1. Node A is searching for content stored at node E. A queries the nearest known nodes to the desired contents ID, which include C. Node C has not stored the desired data and answers in turn with a list of the nearest known nodes, including F, which gets queried next and answers with its nearest known nodes, which include E. Finally A queries E, which returns the desired data. This process is parallelized actually; usually several nodes are queried at once. A more detailed description of Kademlia can be found in the paper by Maymounkov and Mazieres [5].

2) Comademlia: To incorporate compliance management and to uphold the nodes policies at all times several changes had to be made to the routing table and the lookup procedure. One key problem when integrating compliance management into Kademlia routing is the possibility to "eclipse" a node from content tagged with certain tags. For example, if a network has three tags α, β, δ it is possible that the routing table fills, randomly, with nodes that support only α and β. This node now can not find any content tagged with δ, because it has no knowledge of contacts that can handle its desired data. To prevent such an eclipse, Comademlia uses multiple routing tables, where each node maintains one dedicated routing table per tag in its policy. Newly encountered nodes are placed in all routing tables according to the tags in their policies. Furthermore, a node does not maintain a routing table for tags that are not in its policy. This way each routing table represents a unique network view, containing only nodes that are willing to participate for a certain tag. The total size of all routing tables is not a problem for two reasons. Firstly, each table only holds a small subsets of nodes. Secondly, the tables only contain node references that are shared between multiple routing tables of the same node. Furthermore, because a node is most likely present in multiple routing tables, they can be compressed quite efficiently, if needed.

If data is only categorized by a single tag, the lookup works exactly the same as in Kademlia, using the corresponding routing table. However, consider a case where a node has routing tables for tags α containing nodes $\{A, D, F, G\}$ and β containing $\{A, B, D, E\}$ and wants to start a lookup for data that is tagged with both α and β. Respecting the policies of all nodes in the routing tables requires that only $\{A, D\}$ can be used for requesting this content.

More formally, the routing works as follows: let M be the node searching for data, C be the desired data, and C_{tags} a set of tags, which describe the data. Furthermore, let R^M be the set of all routing tables of node M and R_t^M be the routing table for tag t in M. M now first calculates an intersection of routing tables such that:

$$R_{intersection} = \cap_{e \in C_{tags}} R_e^M$$

$R_{intersection}$ can then be used to continue with the Kademlia-lookup, since $R_{intersection}$ only contains nodes that will accept the query regarding this content. The nodes queried in the process then construct the same intersection on their routing tables to determine their list of nearest known nodes in relation to the desired data.

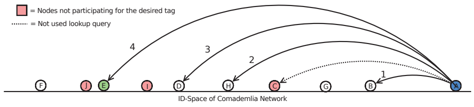

Fig. 2. The Comademlia lookup process for a certain combination of tags.

An example for the lookup process is shown in figure 2. In this example node A computes its $R_{intersection}$, which does not contain C but B. Although C would be nearer to the desired ID, C does not wish to participate in routing for at least one required tag and is thus not part of A's routing table for this specific combination of tags. Instead A queries B with a lookup that contains the desired ID and the accompanying tags of the content. Node B, naturally, only returns nodes which accept all tags for the desired data as well. The lookup process then continues iteratively as is usual with standard Kademlia.

In order to route queries and data each node needs knowledge about the policies of their contacts. A node communicates its current policy to other nodes by adding the policy as often as possible, optimally always, to other messages being sent. Any node can also just query any other node for its policy if needed, e.g. if it is suspected that some policy information might be stale. If the maximum number of tags is a network parameter, the policies can be represented by a bitstring, where every tag has a certain position and can just be set to one or zero, to indicate if a policy accepts the tag or not, respectively.

IV. Evaluation

To test the viability of the prototype we conducted several simulations. In the following, we describe the setup, execution and evaluation of the results in detail.

A. Setup

To simulate our architecture we used the event-based network simulator PeerfactSIM.KOM [10]. We implemented Comademlia as an application for the Kademlia overlay for PeerfactSIM.KOM. All changes to the original Kademlia network, that were outlined in the previous section, were implemented. We also implemented the Kademlia overlay by abiding closely to the original Kademlia paper [5].

B. Simulation Design

We used different simulations to evaluate the behavior and performance of Comademlia. We configured the Comademlia network to distinguish between 50 different tags, since we used real world data from the top 50 tags on the Q&A website stackoverflow.com, which had still significant popularity differences, to model a popularity distribution. Furthermore, all simulations were conducted without churn, since the focus of this work is on the impact that compliance management has on the performance of a peer-to-peer network. Adding churn should not change the relative performance of a peer-to-peer network with compliance management compared to one without compliance management.

For our main simulation we simulated 256 nodes, which are organized in 50 ”groups”, which aggregate nodes that have similar policies. Every group was assigned a uniformly distributed number of nodes between one and 10. We also created 50 different pieces of data which were to be distributed and looked up inside the Comademlia network during the simulation. To configure the main simulation as closely to a real scenario as possible we used external data to model the policy and tag distributions. Tag distribution in a real application will follow some kind of popularity model, with the most popular tag used quite often and the least popular tag used very sparingly. To get real life data, we queried the amount of tags each question on stackoverflow.com gets assigned and how the tags are distributed overall. We found that the number of tags per question is Gaussian distributed with a mean around three. Minimum and maximum number of tags where one and five respectively, since those are a hard cap on stackoverflow.com.

For the overall popularity distribution we gathered the total amount of usages for the top 50 tags on stackoverflow.com. The numbers can be found in figure 3. We used the inverse of the same overall popularity distribution to determine the policies of the node groups. This is sensible because typically very popular content will be accepted by the majority of nodes, while unpopular content probably will be rejected by a large part. The nodes were configured to accept a random number of tags, which was Gaussian distributed with a mean of 40 and a standard deviation of three.

We ran the main simulation ten times with different seeds for 24 hours of simulated time. For every run-through the nodes first joined the network without conducting any other actions, besides determining their policy. After all nodes joined the network, a random node with matching policy was being assigned as the owner of one of the data-pieces. This node tried to store the data inside the Comademlia network. The procedure was repeated until all data-pieces were assigned an owner once. Please note that the owner of the data did not

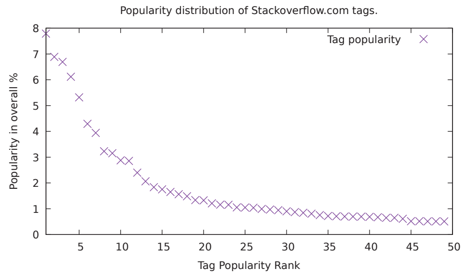

Fig. 3. Popularity tag distribution according to stackoverflow.com tags.

store the data itself, unless its ID was close enough according to the storage algorithm. Following the storage of the data, the retrieval phase starts performing one action per simulated minute and continues until the simulation is finished. During every retrieval phase action, nodes from a randomly selected node group, whose policy allows for it, try to retrieve a random, existing piece of data from the network.

C. Evaluation Results

In this section we present and discuss the results from the main simulation and some follow-up simulations, which we conducted to answer questions brought up by the initial results. Further results can be found at a dedicated web-page [8] online.

1) Accessibility: One of the measured metrics was the accessibility, meaning the number of lookup requests that were successful when searching for existing data. For nine out of the ten seeds, simulated during the main simulation, the accessibility was a full 100%, meaning every node that sent a lookup request for some existing value received a valid response with the desired data. In the remaining case 6842 out of 7026 (97,4%) sent lookups were successful. We investigated what lead to the non-perfect retention rates in some edge cases, and can conclude that it is caused by a network partition for certain tags.

The network partition is a residue effect from the Kademlia underlay and how the system builds and maintains its routing tables. Since the nodes that are added initially to a routing table are dependent on the bootstrap contact, separate networks can sometimes be formed for certain tags. Solving this partition problem could possibly be done by exploiting node lookups, that are not possible in the partitioned view, but can be made through contacts in other routing tables. After a partition is detected, the node can try to start node lookups in an area of the network where the nodes knowledge is not extensive enough. Since the node lookups are not constrained by policies, more nodes can be used for the lookup and potentially help resolve the partition.

Carrying on, we were interested in whether the accessibility rates depend on the number of accepted tags per node. To test the effect of node preferences on accessibility, we started

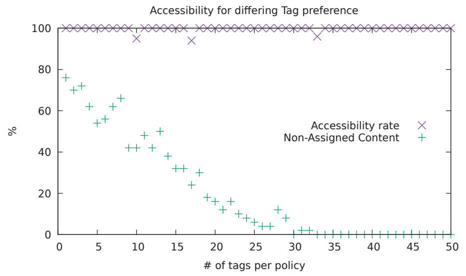

Fig. 4. The accessibility rates and percentage of content that could not be distributed inside the network due to no matching nodes for a series of simulations varying the number of tags accepted by the nodes.

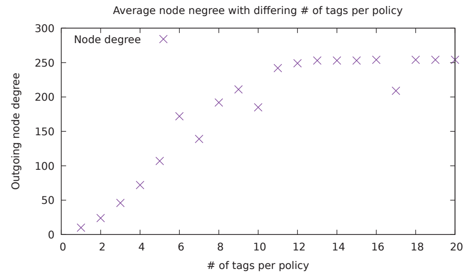

Fig. 5. The outgoing node degree for simulations using different numbers of allowed tags in the nodes policies.

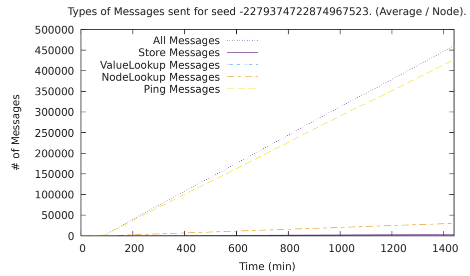

Fig. 6. The different types of Messages for one seed of the main simulation. Response type messages are not shown, since they are just mirrored in type.

a series of additional simulations, which were similar to the main simulation. In every simulation of the series the nodes policies had a fixed number of accepted tags. Starting from one and being incremented with every simulation. Additionally, we only simulated five instead of 24 hours, which was sufficient to compare accessibility. The results can be found in figure 4. As shown the accessibility is generally not influenced by the number of tags that are accepted by the participating nodes. However, there is data generated by the simulation, according to the same rules of the main simulation, that can not be stored inside the network, due to the fact that there is not a single node whose policy matches the tags on the data. For example if every node only accepts one tag, almost 80% of all generated data can not be placed inside the network. This is not very surprising, since the data we generate has on average three tags assigned to it. This further means that, depending on the distribution of tags on content, there are certain thresholds, which show how many tags have to be accepted by nodes on average to guarantee that the network can handle most data. For example, to be able to store about half of all possible data-pieces, the network in our simulations needs to have nodes that accept about six tags on average. To store about 80% of all possible content the nodes need to accept about 20 tags on average. Please note, that this are only the thresholds for this specific type of data. Other data with a different tag-distribution would produce other thresholds.

2) Network Complexity: We also measured the outgoing node degree of every node by taking all active, unique contacts from every routing table into account. The average node degree at the end of the simulations was always near the count of all nodes in the simulation. To confirm that this is not the default case for Comademlia we conducted further simulations. Our hypothesis was, that the node degree is dependent on the number of accepted tags per node, because every tag introduces an additional routing table, which heightens the number of contacts a node can potentially store. We compiled the average node degrees for the tag preference simulations from the previous section into figure 5. The results show, that the outgoing node degree scales linearly with the number of accepted tags per node. The results from the main simulation can thus be explained by the, in comparison to the node count, high number of allowed tags per policy.

3) Message Count: As a last metric we recorded the amount and type of sent messages during the simulation, which are shown in figure 6 for one of the seeds. Overall in the 24 hour simulated period the network generated 117,754,211 messages. The types of messages were very similar for all seeds and distributed (rounded) as follows: 0.7% store requests, 0.01% value lookups, 6.5% node lookups and 92.7% ping messages, which include requests for policy. We also tracked message responses separately, which trivially just mirrored the distribution of the messages. As a comparison we ran a Kademlia simulation which was kept as close to the main simulation's configuration as possible, which was done by omitting all policy and tag information[1]. The results show that the messages are comprised of 7.2% store requests, 0,2% value lookups, 72,4% node lookups and 20,2% ping messages. In sum over the 24 hour period the network generated 9,753,873 messages.

The two main differences in Comademlia are the much higher number of messages overall and how they are distributed. The higher number of overall messages is mainly due to the fact, that not only one, but multiple routing tables are running their maintenance algorithms. One of those algorithm is the automatic refresh. The refresh interval designates how long a bucket of a routing table may be unused for any operation until it is actively refreshed by querying a random node that would fall in its prefix range. The count of ping messages heightens with every routing table as well due to the way contacts are added to the routing

[1] The full results can also be found online at the results page [8]

TABLE I
MESSAGES PER LOOKUP DATA FOR MAIN SIMULATION

Seed	Messages Total	per Lookup
-2279374722874967523	11333	1.61
-3478172347484860844	10995	1.56
-4075461766863512380	10799	1.54
-6967349957205617419	11097	1.58
-7863032748618955690	10709	1.52
3426883308801768639	12698	1.81
5661528113092291996	10845	1.54
6294736137111301708	11733	1.67
6580435484139318149	11190	1.59
8678612569865056655	10678	1.52
Overall	112077	1.60

tables. When a new contact is added to a routing table, some of the established contacts are pinged to check whether they are still alive and cooperating. The smaller number of store requests in the Comademlia network is explained by the fact that in a Kademlia network there are more potential nodes willing to store data, because Kademlia nodes are not restricted by policies as in Comademlia.

To complete the analysis of the message throughput we took a look, at how many messages are generated for a single content lookup. Every seed of the main simulation created 7026 value lookups. In table I the total and average amount of queries per lookup for the different seeds is shown. The table does not take responses into account, which are symmetrical. The overall average of 1.6 queries per lookup can be considered very good, since one query is the least possible amount for a lookup, which can only be achieved if every node knows every target node directly.

V. CONCLUSION AND OUTLOOK

In this paper we introduced the concept of compliance management in peer-to-peer networks. To get an understanding of the new type of network, we discussed its desired capabilities and goals. Based on those assumptions we proposed an architecture, which relies on categorized content and policy based routing. To answer the question whether such an architecture is feasible, we implemented a proof-of-concept inside an event-based network simulator and conducted several simulations. The outcomes of the simulations were evaluated and provided us with insight about how to achieve a better performance in the future. Furthermore, the evaluation showed that compliance management through policy based routing and storage in peer-to-peer networks is a feasible approach, and should be further investigated.

Building on this work, there are several things that have to be researched further. Firstly, there is no notion what optimal performance looks like for a compliance management network. Our evaluations only show us that such a network can function and some notions how to improve, but not how well it functions compared to a hypothetical optimum. In future research we will concentrate on finding a performance optimum for compliance management, which then can serve as a benchmark for future architectures. In its current form Comademlia nodes define their policies as a whitelist, meaning they explicitly state the tags which are accepted. Some research on the implications of a blacklist approach, if any, would be insightful. Furthermore, we will research different possibilities of integrating a fully functioning content tagging system into compliance management architectures, since tags are a crucial part of a successfully operating compliance management architecture, besides policy based routing and storage. Part of this research would not only be the question how to assign tags correctly, but also how to organize the tags between themselves. Comademlia uses an approach where the tags have no special structure, but other approaches, e.g. ontology-like structures, could possibly be utilized to enhance performance.

REFERENCES

[1] J. F. da Silva, L. P. Gaspary, M. P. Barcellos, and A. Detsch. Policy-based access control in peer-to-peer grid systems. In *Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing*, pages 107–113. IEEE Computer Society, 2005.

[2] O. Görlitz, S. Sizov, and S. Staab. Pints: peer-to-peer infrastructure for tagging systems. In *IPTPS*, page 19. Citeseer, 2008.

[3] O. Görlitz, S. Sizov, and S. Staab. Tagster-tagging-based distributed content sharing. In *European Semantic Web Conference*, pages 807–811. Springer, 2008.

[4] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, et al. Oceanstore: An architecture for global-scale persistent storage. *ACM Sigplan Notices*, 35(11):190–201, 2000.

[5] P. Maymounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems*, pages 53–65. Springer, 2002.

[6] A. Muthitacharoen, R. Morris, T. M. Gil, and B. Chen. Ivy: A read/write peer-to-peer file system. *ACM SIGOPS Operating Systems Review*, 36(SI):31–44, 2002.

[7] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*, pages 329–350. Springer, 2001.

[8] A. Schneider. Simulation results for comademlia https://www.uni-duesseldorf.de/redaktion/index.php?id=102391, 2016.

[9] D. Smetters and V. Jacobson. Securing network content. Technical report, Citeseer, 2009.

[10] D. Stingl, C. Gross, J. Rückert, L. Nobach, A. Kovacevic, and R. Steinmetz. Peerfactsim. kom: A simulation framework for peer-to-peer systems. In *High Performance Computing and Simulation (HPCS), 2011 International Conference on*, pages 577–584. IEEE, 2011.

[11] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review*, 31(4):149–160, 2001.

Chapter 4

Reusable Statements in Dialog-Based Argumentation Systems

This chapter summarizes the contributions of the paper [36]:

Alexander Schneider and Christian Meter:
Reusable Statements in Dialog-Based Argumentation Systems
In Proceedings of 1st Workshop on Advances In Argumentation In Artificial Intelligence (AI^3 @ AI*IA), Bari, Italy, November 2017.
Acceptance ratio: 65%.

Online argumentation makes up a big part of the internet. Discussions are found online at almost every news-provider, private blog, forum or social media page. Often those discussions face a few problems. One is their unstructured form. People just write their opinion one after another or sometimes by referring to a singular previous post. A second problem is that the discussions are often repetitions of the same discussion between different platforms. A user, for example, can discuss the Brexit in the comments section of a blog post about the Brexit, then switch to his personal Facebook-page and start a discussion about the Brexit with some of his friends and ultimately read an online news-bit about the Brexit to start the discussion anew in its comments section. In all instances the user is probably going to repeat arguments, face the same trains of thought and wish they could just use the previous discussions to their benefit.

This work discusses such situations to raise awareness of the existing problems, possible solutions and the challenges that need to be overcome to attain acceptable solutions.

4.1 Paper Summary

Here we present the idea of federated argument aggregators that form a distributed, global argumentation graph. The key contributions of are as follows:

1. Bringing attention to the need for a decentralized, connected web of arguments with a technical solution, which allows every argument aggregator to network with peers for the

purpose of exchanging and updating arguments.

2. Introducing and elaborating a set of challenges that need to be tackled to make a distributed argumentation network possible. We also sketch out potential solutions to the defined challenges.

The paper thus lays the groundwork for the dissertation as well as for EDEN. We argue that a decentralized, shared argument network leads to reuse of arguments, because similar discussions happen on different platforms simultaneously. The thought of statements and their relations as shareable and reusable entities is a very important one for the rest of the paper. Gaining the ability to recycle arguments enables the user to spend less time formulating their thoughts into an argument, and instead allows to focus on parts of the discussion that were not yet tackled as extensively. It is reasonable to assume that well-thought-out and enticing arguments will be shared more often, making them more ubiquitous. This also solves the bootstrap problem of discussions. An empty discussion is not inviting to the first users. Instead of having a multitude of cold starts, there needs to be only one if the systems share the arguments belonging to the same topic.

To make such a network function satisfactorily, we need to overcome several challenges. Arguments, for example, are not always context-free. In some cases an arguments' validity depends on the context of the discussion. Consider the argument "It is not a good idea to get a family dog, because no family member has time to walk the dog". This argument may be true and make sense in the context of one families discussion, but be completely invalid in the context of another family's discussion about acquiring a pet. It also would be completely nonsensical in a discussion about whether dogs are better pets than cats conducted on a pet enthusiast forum. One of the proposed solutions is to utilize a *wisdom of the crowd* approach where users of the system vote on the validity of automatically shared arguments.

Another presented challenge is updating distributed arguments. If an argument aggregator updates one of its entities, the changes should be propagated to every other instance that shares the same argument. But this would in turn mean that arguments can be changed on remote systems, which is not desirable. The other two described challenges are of a technical nature, which are the design of a user-friendly interface and a suitable architecture that allows the network to have a good performance while incorporating all necessary functions.

To round out the paper, an architecture sketch is given which focuses on solving the previously defined challenges. Modules are used as the basic building block of the architecture sketch. The proposed modules are responsible for the database, the user-interface, the discussion logic and the sharing of arguments. This has the benefit that not every participant of the network needs to follow the same reference implementation, but instead can exchange any module that better suits their needs. Generally, a federated structure between the members of the network is suggested, essentially building a network of arguments that is akin to the World-Wide-Web.

4.2 Contribution

The main contribution of this paper is establishing the concept of a decentralized network of arguments. To our knowledge there have been no prior scientific aspirations of that exact nature. Other key contributions include the presentation of challenges that a decentralized argument network must overcome. Additionally, we presented one or more possible solutions to the challenges. Most of those solutions have been used in practice in later implementations of a system that enables a decentralized argument network. As a third contribution, we presented a possible prototype-architecture which, as well, was used as a guideline in the development of the subsequently released EDEN.

4.3 Personal Contribution

The author of this thesis, Alexander Schneider, developed the initial idea of a decentralized network of arguments. Core challenges were discussed and expanded upon in close cooperation between the authors, as was the development of resulting architecture sketches. Arguments for the importance of an argument network, the networking based challenges and the challenges regarding the context-dependence have been put in writing mainly by Alexander. The challenges regarding user-friendly interfaces and update-ability of arguments have been mainly formulated in text by Christian Meter.

4.4 Importance and Impact on the Thesis

This paper discusses for the first time the idea of a decentralized argument network, which was the precursor for EDEN. The importance of distributed arguments which was explained in this paper showcased the advantages such a system has in contrast to isolated argumentation systems. The defined challenges were an important step in the design of the EDEN system as they showed what features needed to be especially handled with additional thought. Resulting from the challenges, an architecture sketch was provided, which in major parts was used for the construction of the final system. Summing up, this paper laid the ground-work for the future developments of (decentralized) argumentation systems developed by the author.

Reusable Statements in Dialog-Based Argumentation Systems

Alexander Schneider and Christian Meter

Heinrich Heine Universität Düsseldorf, Universitätsstr. 1, Düsseldorf 40235, Germany,
firstname.lastname@hhu.de,
WWW home page: http://cn.hhu.de

Abstract. Discussions on the Internet are usually conducted in isolation on a single platform, although there are many discussions on the same topic going on simultaneously all over the Internet. We argue that it is possible to connect similar discussions by reusing arguments, thus gaining a connected network of statements, supports and counterarguments which helps eradicate redundant and repetitive parts of common discussions. To achieve this goal we outline challenges that need to be solved and propose a possible architecture to tackle those challenges.

Keywords: dialog-based argumentation, arguments, statement reusability, argument networks

1 Introduction

Nowadays a lot of discussions are conducted online on social media, webpages of news outlets and forums. Those discussions are often unstructured and become hard to follow after they reach a certain size. Dialog-based argumentation systems like D-BAS [5] allow the user to formulate arguments while conducting a conversation with the system. A user can utilize any arguments that other participants of the discussion contributed to deliberate and express her opinion. As field tests of D-BAS have shown, more people participate when they can reuse arguments made by other participants compared to when they are required to formulate their own thoughts into a formal argument. The flaw with such dialog-based discussions is that they are localized and users thus can only re-use arguments made in their specific instance of the system. To solve this issue we propose an architecture to network several discussion and content providers, which host dialog based discussions. The goal is to generate the possibility of (automatic) argument exchange between those providers thus generating a network of reusable arguments and later on whole discussions. The thought of arguments as a persistent reusable resource which can be improved as time goes on is quite compelling. To achieve this goal it is imperative to design and implement this argument network in a fashion which does not appeal solely to argumentation experts, but rather to the general public and the content providers. Since such a system heavily relies on being widely distributed and being used by a lot of people that in turn create arguments, it is the foremost goal to design the system in a fashion which is suited for this target group.

This paper has a twofold purpose. The first is to argue for – and bring attention to – the importance and possibility of an interconnected argument network which can be widely used and distributed. The second is to raise awareness of the specific challenges arising when dealing with arguments which are distributed over several systems.

The remainder is structured as follows. We give an overview on related work in Sect. 2 followed by an outline on the importance of distributing and reusing arguments in Sect. 3. Following, we discuss open challenges for such a system in Sect. 4 and propose a possible architecture in Sect. 5 before concluding the paper in Sect. 6.

2 Related Work

There are a few papers about a system for storing and reusing arguments called "The Argument Web" [1, 2]. The main difference to our proposal is that the Argument Web aims mainly at storing discussions in databases for later uses by a multitude of tools, while we aim to actively distribute and propagate user-generated arguments to be used by other non-expert users in a dynamic network. Heras et al. [3] have researched the formalization of user-generated argumentation on social networks. While we also work with user generated arguments, we go the opposite way and require user-interfaces that facilitate the arguments to be input in an already formalized structure albeit being natural language as proposed by Meter et al. [6]. Similarly Toni and Torroni [4] researched a methodology to convert user-generated comments into arguments.

3 Importance of Distributed Arguments

Reuse of arguments in a dialog-based discussion could help the users deliberate more efficiently. The user can recycle arguments already made by others or be confronted with their opinion on a matter without the strain of necessarily formulating ones thoughts into a formal argument. Since discussions on the Internet are not carried out by experts in the field of argumentation, the quality of arguments varies considerably. Well written and structured arguments would probably be propagated more often and as such heighten the quality of future discussions. Furthermore, a lot of discussions on the same topic happen in parallel on the Internet. As an example, in 2016 there is a high number of discussions about the "Brexit" going on, since every news outlet published stories about it and most of them also allowed discussions on the articles of some sort. Factor in more private discussions on social networks, like Facebook, and the number grows even higher. All of those discussions contain numerous arguments and trains of thoughts that were already stated in another similar discussion somewhere else. If those discussions were at least partially linked, one probably would not see the necessity to restate the same opinions, but would just express their view by agreeing or disagreeing with the available statements or by reusing them in a new discussion. An as of yet untested but likely side-effect of this recycling could be that the users would reach a point where they can continue with a branch of the discussion which is "new" and produces original arguments and statements faster than without recycling.

Another advantage of an argument network would be that new discussions would not have to start empty, since they could be seeded by already ongoing arguments to similar discussions or whole parts of the same discussion at another argument provider. The content providers hosting the discussions would benefit as well, since arguments made on their platform and shared could contain a reference to the place of origin in turn incite traffic to the content providers and argument hosts.

4 Current Challenges

To distribute arguments, one faces unique challenges which are not encountered when dealing with arguments as a single entity belonging to one specific discussion. This section tries to describe the challenges that need solving to fully realized distributed arguments in a real world setting outside of academia.

Development of a Distributed Architecture. Naturally, for arguments to be distributed there has to be the technical foundation allowing content and argument providers to store arguments and subsequently share them. All possible architectures have to be performant enough to support a large number of providers sharing arguments simultaneously. We acknowledge that this challenge is more geared towards the networking community, but want to emphasize its importance nonetheless. We furthermore provide a sketch of a possible architecture in Sect. 5.

User-Friendliness. A system relying on the participants to reuse arguments has to provide the right tools making it as easy as possible for the participant. One example could be a kind of universal bookmarks. E.g. if a user participates in a discussion on news-outlet X and sees a clever argument that she likes, she should be able to mark it for future use during a discussion on any platforms Y and Z. Optimally this should be hardware independent so the user can fluently switch between devices. Another possible helper for reusing arguments could be a service which suggests existing arguments of other platforms while the user is typing. Although, this solution requires a knowledge of most arguments in the network, which could turn out as an impossible task to solve efficiently.

Update of Arguments. In a system where arguments propagate between different systems and hosts, there is also the problem of how to handle updated arguments. In a user-driven system arguments are subject to change because of spelling or grammatical errors. These changes should optimally propagate to all systems reusing said argument. If and how this happens depends mainly on the architecture. From a networking view the choices are to build a highly interconnected network where updates are distributed as widely as possible but require a structured network that needs to be maintained. The other end of the spectrum is a loosely related network of federated hosts that exchange updates at will. This solution has a low overhead but also does not necessarily distribute all updates. In our architecture sketch we use a federated network, which uses a subscription system for arguments and topics to receive updates. Furthermore, the community of an argument host can be allowed to curate the acceptance or rejection of propagated changes as the system is mainly user-driven.

Context-Dependence of Arguments. Ideally, we do not only want to reuse arguments but also automatically import all supports and attacks of a reused argument as this would deepen the discussion without any effort at all. The problem here is that some arguments posses a context, which makes it impossible to import more than the argument itself. For example in a discussion about raising the quality of life in a town with little money, there could be the argument A "Lets build a park, since it raises the quality of life". An attack B on this argument could be "A park is too expensive for the current town budget". Now there is a similar discussion going on in a more wealthy town. Some participant reuses A, because she finds it a compelling argument. If B is imported automatically as well, it does not fit because the context of the town in question having a tight budget does not apply. Possible solutions for this problem can be found with natural language processing techniques that try to determine whether statements posses context or are context-free. Another possible solution would be to allow the participant that imports the argument to choose whether attacks or supports shall be imported as well. Although this could have an adverse effect on the participation rates, since it heightens the amount of work for the participant.

5 Architecture Sketch

A possible architecture for a distributed argument network should consists of interchangeable parts or modules to accommodate the heterogeneous requirements of different content providers. The modules need to be exchangeable as long as they fulfill a certain set of requirements. The main modules we propose are the user interface, the execution logic engine, the database, and a module which we call the *aggregator*. The database is used for plain storage of arguments that a host collected over its lifetime. The database in turn connects to the aggregator, which has a multitude of tasks. The most important task of the aggregator is to communicate with the aggregators of other hosts to exchange arguments when needed and also tend to fetching and retrieving updates on existing arguments. For faster access the aggregator should also provide a cache of the most used arguments, to be able to quickly answer queries without the need to communicate with the database too often. Furthermore, the aggregator coordinates information flow between the user interface and the execution logic. When a user interacts with the system through the user interface, the provided data is forwarded to the aggregator which provides additional arguments if needed and queries the execution logic engine for the next steps before sending the result back to the user interface. As such the aggregator is the communication hub in the envisioned architecture. An explanation on how the execution logic engine works is out of scope for this paper, but can be found in detail in the D-BAS paper [5].

In general, the network that would form between discussion hosts would be a federated network, imitating the Web. A provider of content that is willing to host discussions can deploy an implementation of the proposed architecture. After that the different hosts start to connect loosely every time arguments are exchanged between them. The first exchanges are initiated through users recycling arguments they have seen on other hosts. This is the exact reason why the system needs to give a user the capability to "bookmark" arguments. Hosts that know each other can establish a more solid relationship by interchanging arguments based on set rules instead of on demand by users. Much as the web, a federation of every willing provider should be possible, regardless of the size or power of the provider. Whether a private web-blog or a huge media outlet or a social media network decides to provide an argument host should make no difference on the network and the users.

6 Conclusion

In this paper we argued for the need of a system that facilitates reuse of (user-generated) arguments and discussions. We emphasized the benefits of such a system and pointed out big challenges which need to be solved before putting such a system in place. We also provided the sketch of an architecture for such a system. The proposed architecture utilizes a federated network of content-providers which share user-generated arguments and discussions. For future research on this matter an enhanced prototype implementation of the proposed architecture incorporating as many solutions to the open challenges as possible offers itself up.

References

1. Bex, F., et al.: "Implementing the argument web." Communications of the ACM 56.10 (2013): 66–73.
2. Rahwan, I., et al.: "Laying the foundations for a world wide argument web." Artificial intelligence 171.10-15 (2007): 897–921.
3. Heras, S., et al.: "How Argumentation can Enhance Dialogues in Social Networks." COMMA. (2010): 267–274.
4. Toni, F., Torroni, P.: "Bottom-Up Argumentation." TAFA 7132 (2011): 249–262.
5. Krauthoff, T., et al.: "Dialog-Based Online Argumentation." COMMA. (2016).
6. Meter, C., et al.: "discuss: Embedding Dialog-Based Discussions into Websites." International Conference on Learning and Collaboration Technologies. Springer. (2017): 449–460.

Chapter 5

EDEN: Extensible Discussion Entity Network

This chapter summarizes the contributions of the paper [26]:

Christian Meter, Alexander Schneider, and Martin Mauve.
EDEN: Extensible Discussion Entity Network.
In Proceedings of Computational Models of Argument (COMMA),
Warsaw, Poland, September 2018.
Acceptance ratio: 50%.

Many web services allow users to discuss a multitude of issues. But even if the discussions are centered on the same issue, there is currently no easy way for the discussions to be merged and for arguments used in them to be reused elsewhere. This is why, based on the ideas formulated in Chapter 4, we developed a system which allows the hosts of discussions to establish links between them in a federated manner. Those links allow users and hosts to reuse and to exchange statements and their interrelations. This allows them to treat arguments as a commodity, which can be freely shared, in turn creating more extensive discussions and argumentation data.

Our implementation takes into account technical challenges, that are described and solved in this paper, as well as requirements of the users and the hosts that need to be incorporated. The requirements are of special importance, since the goal of the application is to be used in real-world discussions and not solely as a scientific prototype.

5.1 Paper Summary

This paper presents two main contributions, which are as follows:

1. An architecture, which allows for the decentralized distribution and revision of argumentative statements and their interrelations.

2. A reference software implementation based on the introduced architecture.

This paper is the core-piece of the dissertation. All previous work led to the creation of a decentralized argument network, and all works thereafter are based heavily on the existence of EDEN. In the paper we introduce the notion of a decentralized argument network before we review related work in the fields of argument storage and reuse. Then, we specifically define an environment which is needed for distributed arguments to be a valuable and reusable asset. The focus therein lies mainly on websites that create and curate content, which we name *argument aggregators*, and their differing policies regarding acceptable content.

Following the definition of the environment it is important to define an architecture to manage distributed arguments in said environment. Two data-structures, statements and links, are introduced to be used as atomic building blocks in the system to be. Those structures allow decentralized versioning, deletion, and modification. To provide examples, we define all possible modes of updates and modifications and how they would be executed in the proposed architecture.

The last major section introduces our software implementation of the previously described architecture: EDEN. EDEN is based on functional modules, which define clear boundaries as to be kept interchangeable. This allows every participant of the network to replace any module if needed. The instances of EDEN automatically exchange relevant statements and links with known entities that are not blacklisted or otherwise untrustworthy. To round out the section we discuss possible challenges with such a system and how EDEN tackles them.

We conclude the paper with a description of further optimizations that can be pursued as well as the "hands-on" experience which we collected with the system prior to the publication of the paper.

5.2 Contribution

This papers' main contribution is the presentation of an architecture which can handle distributed, versioned and reusable arguments. This is the single most important contribution of this thesis, as it hopefully enables countless future applications that can build on the paradigm of distributed arguments. The advantages of such an architecture are – among others – fewer filter-bubbles, less clutter in discussions and deeper argumentation graphs. Other contributions include a modular open-source implementation of the architecture, as well as the presentation of preliminary experience with the architecture.

5.3 Personal Contribution

Alexander and Christian Meter developed the architecture presented in this paper in equal parts. Implementation of EDEN was also done cooperatively, whereby Christian Meter contributed one third of the software and Alexander Schneider the rest. The paper was written in equal parts by Alexander Schneider and Christian Meter with editorial help and fruitful discussions contributed by Martin Mauve.

5.4 Importance and Impact on the Thesis

This paper is the core contribution of the thesis. By introducing an architecture and prototype for the previously published idea of a web of decentralized arguments we laid the groundwork for all our future contributions in the space of decentralized argumentation and argumentation software. We use EDEN in following field experiments and evaluations as a back-end for all other discussion software. This shows that decentralized argument storage and shared arguments between systems are ripe for real-world use. On account of the paper results, we do not use unconnected arguments in our work anymore.

September 2018

EDEN: Extensible Discussion Entity Network[1]

Christian METER, Alexander SCHNEIDER [2] and Martin MAUVE
Computer Networks Department
Heinrich-Heine University Düsseldorf, Germany
firstname.lastname@hhu.de

Abstract. Enabling the reuse of arguments as entities that can be shared across multiple Internet-based discussion platforms and that can be improved upon while they are being used and reused has many benefits ranging from easier participation in an online discussion to increasing the quality of arguments. In this paper we propose a mechanism that is able to support the large-scale reuse of arguments by providing distributed version control of argument data. Building on that mechanism we have designed and implemented *EDEN*, a framework which enables platform providers to easily network their discussions. EDEN is designed for real-world use and provides all tools necessary to enable the reuse of arguments and their interrelation for users and providers alike.

Keywords. massive online discussion, discussion networks, EDEN, discussion graphs

1. Introduction

Arguments and their interrelation are valuable resources. They require effort to craft and they reflect the knowledge and opinions of those that have contributed them. Furthermore, their value grows as a network of arguments and their interrelations increases in size. On the Internet this is currently not supported in an appropriate way. Most arguments are ephemeral postings in forums and comment sections of news media. Even dedicated argumentation websites do not allow for connecting arguments across multiple websites. In order to address this problem Bex, Lawrence, Snaith and Reed have introduced the notion of an *Argument Web* [6]. Unfortunately, the Argument Web has not (yet) gained sufficient traction and is limited to a set of research prototypes.

In this paper we argue that in order to have a larger impact, the Argument Web needs to be more than a way to specify, describe and reference arguments. In particular it has to take into account the specific needs of those that operate websites where argumentation is taking place and of the users that visit those web sites. As we shall discuss, this leads to a challenge on the system level that can be summarized in a simple question: How can arguments and their interrelations be managed as persistent resources in a distributed

[1] Submitted to IAT.
[2] Both first authors contributed in equal parts to this work.

(web-based) system? As an answer to this question we propose the idea of an *Extensible Discussion Entity Network* (EDEN).

EDEN is designed to provide persistent arguments and interrelations between them, which can be shared and reused, while incorporating the manifold requirements of users and platform providers. Those, sometimes contradicting, needs are usually not considered, when designing systems in the argumentation space. We believe that EDEN facilitates adoption for real-world scenarios.

The goals of this paper are twofold. First we would like to raise awareness for the fact that there are system-level challenges that need to be addressed in order to make the idea of an Argument Web work in a real-world setting. Second, we present a solution for the most important of those challenges, namely how to distribute and manage argument data in an heterogeneous Internet-based environment.

The remainder of the paper is structured as follows. Section 2 examines related work and compares it to our contribution. Next, we introduce our view on distributed argumentation and its stakeholders in Section 3. Section 4 then discusses a method for versioning arguments in a distributed environment. Following that, we present an implementation of EDEN which describes its functionality and specifics in Section 5. Finally, in Section 6 we conclude the paper with a summary and an outlook on future work.

2. Related Work

The idea of a connected network of arguments is not entirely new. The general idea for an "Argument Web" was established by Rahwan et al. [11] and further refined by Bex et al. [6]. Following the general idea a central database for the Argument Web was created by Lawrence et al. [8] which in turn interoperated with different applications belonging to the Argument Web [5,7]. The point where EDEN differs from that work is that we do not utilize a central database, which acts as a central interface for import and export for arguments in the AIF format. Instead we aim for dynamic exchange in a federated network of providers. Furthermore, EDEN is not bound to any special ontology, but instead focuses on arbitrary "atomic" entities.

There is also work by Rowe et al. [12] where the concept of reuse is anticipated by designing a system where it is possible to import and export arguments into and out of the Araucaria system on local instances.

Argument reuse has also been touched upon outside of the argumentation community. Kelly et al. [14] proposed the reuse of arguments via design patterns to ease the construction of safety cases and Smith and Harrison [13] proposed a system for reuse of descriptive arguments in hazard classification. To our knowledge EDEN is the first system to aim for reuse of arguments made by layman in a distributed online argumentation environment.

3. Reusable Arguments and their Environment

In order to be able to tackle the systems-level challenges posed by the idea of reusable arguments, we need a good understanding of the environment, where those arguments are created and (re)used. This environment consists of websites and web-based services that host discussions. In particular, this includes online newsmedia, social networks and discussion forums. We term these websites and web-based services *argument aggregators*, since they aggregate arguments provided by users in order to form online discussions.

Argument aggregators typically have policies on what an acceptable user-provided argument is and they have mechanisms in place that ensure that the contributions of the users adhere to those policies. The policies of different argument aggregators are quite heterogeneous, thus the same arguments might be acceptable for some content aggregators while others would consider them a violation of their policies. Furthermore, argument aggregators typically perceive the arguments provided by the users as a valuable commodity which helps gain page impressions and generate income, hence they are unlikely to be willing to share them, unless they get something in return like a reference to their web-site or something similar.

Arguments consist of statements that are linked to each other by different types of relations. They are regularly provided by the users of an argument aggregator. Arguments are also often linked to the content of the argument aggregator, e.g. they might pertain to a discussion regarding a blog entry or a news-media article. A specific argument is initially submitted by a single user to a single argument aggregator. However, any user might later on be willing to improve the argument, for example by correcting spelling errors or by making a statement more concise. The users might also want to use a given argument in another discussion, potentially hosted by a different argument aggregator.

Arguments are interconnected. Each argument, potentially, has numerous relations to other arguments. Furthermore, arguments might only be valid in a specific context. I.e., an argument might contain implicit information, that are not specifically stated. For example, the argument “Our labs are in bad shape, therefore we need to invest in new lab equipment.” includes implicit information about the condition of the author’s working environment since not all existing laboratories are in bad shape.

4. Distributed Management of Arguments

The characteristics of argument aggregators, users and arguments lead to challenges at the systems level that need to be addressed in order for the idea of persistent and reusable arguments to come true. The most prominent one is the development of a suitable architecture for the storage and distribution of arguments, where arguments are updated in an appropriate way, if they are used by multiple argument aggregators.

Since argument aggregators are independent entities that desire autonomous control over the arguments they store, show to their users and distribute to other argument aggregators, the architecture of a system for reusable and persistent arguments needs to be distributed. Given that arguments and their interrelations can be modified and improved upon over time, this immediately raises the question how their shared state can be managed.

One option is to take all proposed updates and calculate a resulting state that is then used by every argument aggregator. This, however, entails two problems. First, there needs to be a mechanism calculating a shared global state, which is a hard, but potentially solvable, problem in a distributed system. Second, all argument aggregators would have to agree unanimously on how to handle all updates – in particular whether to accept a given update or reject it. This is unlikely to be feasible in a real world environment.

If updates are optional, however, arguments may have different states at different aggregators. This inconsistent state is likely to cause problems. For example, an attack on an argument may be valid only for a certain variant of that argument that exists only on a subset of providers, since others modified it. Thus it is not clear how the attack can be reused in a distributed environment where aggregators have different versions of the attacked argument.

To solve this dilemma, we propose an approach derived from distributed source code versioning: arguments, or rather the statements and interrelations that make up the arguments, do have a version. An update produces a new version without modifying the original. The updated version refers to the original(s) as it's predecessor(s), effectively preserving history. This allows both for persistence, since no version is ever deleted and free choice of the content aggregators regarding what updates to accept.

In order to support distributed versioning of arguments, two problems have to be addressed. On the one hand, appropriate data structures are required that support the versioning of arguments. On the other hand there needs to be a mechanism to distribute information about new versions to those that might be interested in updates.

4.1. Data Structures for Versioning Arguments

In order to provide versioning for arguments we first determine the entities that make up a network of arguments. Those are statements and relations between statements. We then define an object to be a specific version of a specific entity. The data structures for storing objects have some common elements for both statement objects and relation objects: a global identifier $\mathscr{N}_{host}$ for the argument aggregator that created the current object (for example, the DNS host name of the argument aggregator), a local identifier $\mathscr{N}_{id}$, that uniquely identifies the entity stored in the object amongst all the entities that this argument aggregator has created objects for, and a version number $\mathscr{N}_{version}$ that indicates a specific version of the entity at this argument aggregator. Together those three values represent the object-id $\mathscr{N}$ which uniquely identifies a specific object. An important aspect of the object-id is that it can be determined locally and does not have to be coordinated amongst argument aggregators.

Furthermore, each object also has a flag d that indicates if it has been marked for deactivation. The latter is required since nothing should ever truly be deleted when doing versioning. Therefore a deletion of an object is just signaled by a specific version of that object where this flag is set. Providers can chose to follow the deactivation by making the object inaccessible to their users.

In addition to the information that is common to all objects, a statement object contains the following information:

- $\mathscr{P}$: a set of pointers to immediate predecessor versions, which is either a set of object-ids, or $\mathscr{P} = \emptyset$.
- $\mathscr{C}$: the data that makes up the statement, typically a plain text and meta information such as the author of the statement and the authors of modifications to the statement.

Summarizing, a statement-object can be fully described as a tuple $\langle \mathscr{P}, \mathscr{N}, \mathscr{C}, d \rangle$.

Relations include the following additional information:

$\mathscr{S}$: the relation's source, which is the object-id of any statement

$\mathscr{D}$: the destination of the relation, which is the object-id of any statement or another relation

t : the relation-type, e.g. "attack" or "premise-conclusion-relation"

A relation-object is thus described by: $\langle \mathscr{N}, \mathscr{S}, \mathscr{D}, t, d \rangle$. Relations are treated as immutable, they can only be created and deleted, but their content never changes. Therefore they do not need a predecessor.

We do believe that these data structures are sufficiently generic to capture arbitrary argumentation schemes, by utilizing $\mathscr{C}$ as a store for atomic entities of a scheme, and yet they provide all the information required to support versioning. They are also quite easy to extend if the need should arise.

4.2. Versioning Arguments in a Distributed Environment

An object (and thus a specific version of a specific entity) has an *authoritative argument aggregator*. This is the argument aggregator that created it and it can be easily determined by looking at $\mathscr{N}_{host}$ of that object. Another argument aggregator can import that object in order to integrate it into an argumentation that it hosts. After a provider imports an object, it can register with the authoritative aggregator for that object in order to receive updates regarding the entity contained in the object.

When an authoritative content aggregator updates an entity, it creates a new object for the new version of that entity with a new version number. It then notifies the argument aggregators that have registered with it regarding that entity. Those argument aggregators can then choose to accept the update or they can stick with the old version. This is a local decision that could be made by a dedicated moderator, the users of the argument aggregator or by means of a policy where one argument aggregator decides to trust another argument aggregator to provide reasonable updates.

If an entity is updated by an aggregator that is non-authoritative, a *fork* is created. A fork is a new object. For example if the original statement object was $St_1 = \langle \emptyset, id_x, \mathscr{C}, 0 \rangle$ with $id_x = \langle someaggregator.com, 42, 0 \rangle$, then the new fork-object including the new version of that entity could be $St_2 = \langle \{id_x\}, id_y, \mathscr{C}_2, 0 \rangle$ with $id_y = \langle anotheraggregator.org, 13, 0 \rangle$. The aggregator which created the fork is authoritative for that fork. When a fork of a statement is created, all relations belonging to the forked statement are copied and all instances of the forked statement are replaced by the fork in the copied statements. This does not update existing relations, but rather produce new ones specifically for the fork-object.

When an aggregator F creates a fork, it contacts the authoritative aggregator A of the object that was forked. A can decide to ignore the update. Then nothing happens and A remains authoritative for the original object while F is authoritative for the forked object. Or A can accept the update. In that case, it creates a new version of that entity by creating an appropriate object, which has an incremented version-number, updated content and the fork-object as its predecessor, to keep the version history accurate. As with all updates, the new object is then transmitted to all argument aggregators that have registered with the authoritative aggregator regarding that entity. In particular this is received by F. Once F realizes that its update has been accepted, it replaces the fork with the received update.

4.3. Example of Fork and Update Processes

In order to illustrate how the proposed versioning scheme works, we now present an example showcasing the fork and update processes. The example begins as an aggregator with the global identifier *a.com* creates a statement which looks as follows: $S = \langle \emptyset, id_a = \langle a.com, 24, 0\rangle, \mathscr{C}_1, 0\rangle$. Now there are several cases that can occur.

4.3.1. Updating the Statement

Through a user-driven process, *a.com* decides to update the content of the statement S, producing new content $\mathscr{C}_2$. As a consequence an official updated statement-object $\langle \{id_a\}, \langle a.com, 24, 1\rangle, \mathscr{C}_2, 0\rangle$ is created and published to all other aggregators using S. Those aggregators decide individually whether they stick with the old version or update to the new one.

4.3.2. Creating a Fork

An aggregator *b.org* is using S and wants to update the statement's content to $\mathscr{C}_3$. A fork is now created which looks as follows: $\langle \{id_a\}, id_b = \langle b.org, 40, 0\rangle, \mathscr{C}_3, 0\rangle$. This fork is reported to the original aggregator *a.com*. In case *a.com* rejects the update, nothing more happens. If *a.com* accepts the update, it creates an updated version of S and sets the fork as a predecessor to preserve history – resulting in: $\langle \{id_b\}, \langle a.com, 24, 1\rangle, \mathscr{C}_3, 0\rangle$. This is then published to all other aggregators using S. Upon receiving the new object, *b.org* replaces the fork with the update, since its own changes have now been incorporated by *a.com*.

4.3.3. Simultaneous Forks and Updates

Continuing the example in Section 4.3.2, *c.net* is now also using S. It, too, has created an update to S with the content $\mathscr{C}_4$, which results in the object: $\langle \{id_a\}, id_c = \langle c.net, 1337, 0\rangle, \mathscr{C}_4, 0\rangle$. This fork is also communicated to *a.com*, which already updated S after accepting the fork from *b.org*. *a.com* can now choose to incorporate both forks in a new update where the content is then $\mathscr{C}_5$, thus producing $\langle \{id_b, id_c\}, \langle a.com, 24, 2\rangle, \mathscr{C}_5, 0\rangle$. In this version, both the objects from *b.org* and *c.net* are predecessors of the updated object. Figure 1 showcases the relations in this scenario. *a.com* could have also chosen to solely use the fork from *c.net* as the most current version 2, disregarding the changes of *b.org* included in version 1, effectively creating: $\langle \{id_c\}, \langle a.com, st_1, 2\rangle, \mathscr{C}_4, 0\rangle$. Again, this is published to all other aggregators using S. Upon receiving this, *b.org* and *c.net* decide whether they want to stick with their current version or update to the new one.

5. EDEN

This section introduces EDEN, the implementation of the aforementioned ideas. We briefly describe the basic concepts of EDEN before we lay out the modular architecture, several optimizations and first experiences of usage.

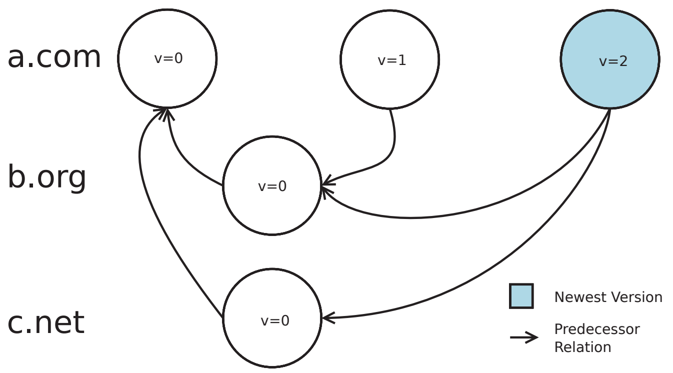

Figure 1. A visualisation of predecessor-relations between different forks and updates of a statement.

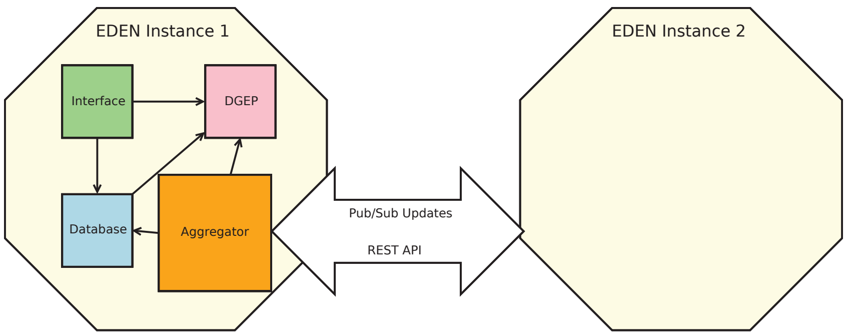

Figure 2. Dataflow between the modules of one EDEN instance. The dataflow with other EDEN instances is established via Pub/Sub and a REST API.

5.1. Basic Concepts

As described in the section above, EDEN is realized as a *federated network* of argument aggregators, where each aggregator is responsible for the state of its own data. Every argument aggregator that wants to enable its community to participate in the global argumentation network, can start up an EDEN instance, which discovers other instances through its initial whitelist and through foreign arguments discovered from those whitelisted instances. The most important task of EDEN is the management and exchange of local and foreign statements and relations. To this end the federated network maintained by EDEN has two logical layers – the local community of an aggregator and the global community spanning all available EDEN instances and their users.

Ideally, EDEN instances should be run by entities which are trusted by their users, like newspaper outlets, NGOs or other organizations. We do not, however, place any firm restrictions on which entities can run an EDEN instance.

We have developed EDEN with modularity in mind. EDEN therefore consists of independent modules, which can be exchanged, as long as they adhere to interface definitions between module "seams". Everything from the aggregator logic, the interface, the database to the execution logic can easily be customized and exchanged in individual EDEN deployments.

5.2. Architecture

The general architecture and dataflow of EDEN's architecture is shown in Figure 2. There are four main modules at work – each with its own purpose.

The interface module enables layman-users to participate in a discussion with their arguments. This in itself is a non-trivial challenge. We use *discuss*, described in Meter et al. [2], as an example implementation of the interface module. In order to allow users to easily import arguments, we present the user with similar arguments from the local and global community, while the user is trying to formulate their own thoughts into an argument by typing parts of it. Similarity here being the analogous and logical proximity of words being typed in respect to potential new arguments. There are many other ways how this support could be realized, e.g. by being able to bookmark arguments at one argument aggregator and then later on reuse these bookmarks in other argumentations at the same or a different argument aggregator. We chose this method to not impose any extra strain on the user in order to not deter them from using the system.

The aggregator module is, metaphorically speaking, the communication central and brain of the operation. All entities at one point pass through the aggregator module. Its duty can be divided into two sections. First, obtaining data from external EDEN instances and providing the local data back to them. Second, coordinating the internal flow of data to make sure it proceeds efficiently between the modules. Our implementation of the aggregator module provides a REST API to enable foreign EDEN instances to query it for data. We furthermore use the *RabbitMQ* publish/subscribe system for queues, to which the aggregators subscribe to be informed about updates to the subscribed entities.

The database module needs to store and efficiently provide heterogeneous data to the other modules. One could use traditional relational databases, but to simplify the storage and query of potentially big amounts of different data-types, EDEN uses an *Elasticsearch* database. One of the many advantages of Elasticsearch is the semantic search, which allows for sophisticated queries, e.g. searching for synonyms. This helps with the provisioning of relevant arguments in respect to the users input.

Finally there is a *Dialogue Game Execution Platform* (DGEP) as defined by Bex et. al. [4]. We use Krauthoff's *Dialog-Based Argumentation System* (D-BAS) [3] for this purpose. The DGEP is responsible for handling all necessary steps in a discussion, utilizing a predefined set of rules applying to a "natural" discourse. Through the modularity any DGEP could replace D-BAS inside the EDEN framework as long as it adheres to the interface conventions between the modules. Currently, the DGEP module also doubles as the module which creates structure data from user input. The choice for using D-BAS in the default version of EDEN is not made because of any architecture considerations, but because we simply needed to pick any one DGEP we could work with to provide a functioning implementation.

The communication with foreign EDEN instances is established in two different ways. If one instance is looking for an entity which may be stored at a different instance, it can query the remote aggregator via a REST API. This will provide it either with a "not-found" answer in case the entity could not be found or with the found entity and a publish/subscribe channel in the successful case. The querying instance can subscribe to the channel if desired to receive updates about new entities or changes in entities, i.e. new versions, thus making the pub/sub system responsible for push-based updates and the REST API for initial queries and pull-based updates.

5.3. Statements and Links

EDEN uses the object types statement and relation[3] as described above. *Statements* are implemented as shown in Listing 1 with some required and some optional keys. The triplet of `[:aggregate-id, :entity-id, :version]` provides a unique address for a specific version of a statement entity. In particular this address can be used by non authoritative argument aggregators to refer to this version.

```
(s/def ::statement
  (s/keys :req [::author ::content ::created
                ::aggregate-id ::entity-id ::version]
          :opt [::ancestor-aggregate-id ::ancestor-entity-id
                ::ancestor-version]))
```

Listing 1: Definition of a statement.

Links are represented as immutable objects, which are defined by a type, source and destination in our implementation. The type represents the relation (e.g. attack, support, undermine, ...) and source and destination are references to objects in a specific version[4]. Since the links are immutable, they can be propagated alongside statements through the pub/sub channels and REST API. The aggregators can then resolve the link-references to the statements and show the users the appropriate versions[5].

```
(s/def ::link
  (s/keys
    :req [::author ::type ::created
          ::from-aggregate-id ::from-entity-id ::from-version
          ::to-aggregate-id ::to-entity-id ::to-version
          ::aggregate-id ::entity-id]))
```

Listing 2: Definition of a link.

5.4. Context Dependent Arguments

To properly import an argument into a foreign discussion, the reused data must be context-free. Our initial approach was to reuse statements and links in a way, that automatically included the reuse of all connected links and statements (e.g. attacks and supports) thus linking both argumentation graphs automatically. This does not always work, since statements may implicitly carry context pertaining to a specific discussion. For example if a family is discussing the acquisition of a pet the statement S_1: "Dogs are good family pets" may be used, with the corresponding attack A_1: "We do not have time to walk a dog every day". The attack is true in the context of the family discussion, because it implicitly carries the information, that the family is too busy to care for a dog. If S_1

[3]In the implementation relations were called links.

[4]The source is always a statement, while destination can be a link or statement.

[5]The current published version has the destination-version as an optional part for a link. This will change, according to the description in Section 4, in the next release.

is now reused in the discussion of an animal-fan forum where the participants want to dedicate a lot of time to their pets and A_1 is automatically presented as an attack, it might not make a lot of sense.

There are different approaches which can be taken to solve this problem. The solution we choose to implement is an “intelligence of the masses” approach. This provides users with the ability to judge about context dependence of automatically imported statements in a review system, before they are fully added and presented to all other users in the discussion. The arguments can be judged one-by-one ordered in a queue accessible to the community members. This works as follows: When a user imports a statement, all other statements which have a relation with it are placed in this new queue. The reviewing users are presented with the statement at the head of the queue, which may be imported if its context-free, as well as with the statement that caused the import of the statement to be judged. The users can then vote to reject or to accept the import. Please note, that the users do not vote on their opinion regarding the content of a statement, but whether the import of it is sensible in the context of the discussion. If a majority of voters accept the import, the statement is fully added to the local discussion and its immediately related statements are placed in the queue. To not overflow the queue with a growing number of review cases, it is capped to a reasonable maximum number of review cases. If the queue is nearing its maximum, statements which are closest to manually imported ones are prioritized. This should prevent the case where one imported statement fills the queue solely with its related statements, while others are left out. The success of this procedure relies on the user’s ability to make objective contributions regarding natural language arguments, which is a feasible assumption as shown in a field study [1] for the D-BAS system, where the users were quite capable in reviewing different aspects of reported statements and arguments. A similar approach to include the community is also heavily used on the StackExchange platforms, e.g. *StackOverflow*[6].

5.5. Further Optimizations

We also implemented some optimizations which help EDEN to better perform its tasks of fostering argument reuse.

We implemented a background entity crawler to optimize argument recommendations to the user. The crawler activates periodically when the instance has unallocated resources and queries foreign instances for yet unknown entities which are then indexed to enhance the lookup-time in the future. The crawler always tries to index the most relevant entities first. In our case this means e.g. statements which are directly – and if none can be found – indirectly related to already known statements. This is done because the chances are higher a user will import statements more closely related to statements already present in the discussion than otherwise. Random entities are queried when all related ones are already indexed.

The aggregator, furthermore, uses a tiered system for retrieval of entities to optimize the information-flow. If it is queried for an entity, the aggregator first attempts a lookup inside its cache. Upon failing to find the desired item in the cache, the lookup is directed to the database. If the entity can not be found in the local database either, it is retrieved from a foreign EDEN instance. This guarantees that the entities are found as fast as possible, since slower queries to the database and to foreign instances are reduced. Of

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

course the last tier of querying remote aggregators is omitted if the query originated from a foreign instance.

5.6. Hands-On Experience

EDEN was written entirely in Clojure and can be freely obtained at github.com[7]. It can be run without further installation from the *Docker* virtual environment, for which we provide the proper configuration. The Docker container also includes a D-BAS and a discuss instance, which are used as DGEP and interface of EDEN, as mentioned in previous Sections.

We conducted first small-scale tests between two and three instances in small mockup-environments running in different Docker containers. Each container was configured to simulate a physical instance on the same network and we used statements and links which were gathered in a field study using D-BAS [1] and split them up into different subsets used by distinct test-instances.

The tests were not meant as definitive performance simulations or a scientific study, but to get an inkling of how multiple EDEN instances behave together. As we expected, the exchange of arguments worked without any further complications and felt natural to the user. Overall the user-experience did not differ from a normal usage of discuss without the EDEN network – except for the larger selection of pre-formulated arguments – which is a positive sign that the user-facing parts are working as intended and do not inherently add any extra strain on the user. Naturally, this was only conducted to gather a general first experience and we will conduct further real-world tests in the future to obtain more scientifically robust data.

6. Conclusion and Future Work

In this paper we introduced EDEN as a framework to enable discussion-entity reuse between different argumentation platforms. We discussed the challenge of keeping a consistent state in a distributed environment and the resulting challenges for versioning arguments. Our work contains solutions for versioning arguments in a distributed network as well as a solution for context-dependence of entities. Furthermore, we introduced a working implementation of the EDEN framework which is open source and freely available to use. The implementation also contains several technical optimizations and performed successfully in first small-scale tests.

One main challenge that remains as future work is the deployment and evaluation of EDEN by real-world argument aggregators. We are currently in the process of negotiating with companies that provide software for online-participation processes such as participatory budgeting and urban planning. We do believe that this might be an excellent starting point for sharing arguments, since there are many distinct online-participation processes that share common topics. Real world adoption could also be furthered by adding DGEP modules for argument aggregation services like www.debatepedia.org or www.procon.org. or by incorporating argument mining modules for unstructured natural language arguments from e.g. social media.

[7] https://github.com/hhucn/eden

We also plan to release improved versions of EDEN. Improvements can be pursued by designing methods to ease the reuse of arguments for the users even further. A shared user-base between different EDEN instances could be pursued to facilitate adoption of the network. Additionally, the technical performance of the framework can be improved upon as well.

References

[1] Krauthoff, T., Meter, C., & Mauve, M. (2017). *Dialog-Based Online Argumentation: Findings from a Field Experiment.* Proceedings of the 1st Workshop on Advances In Argumentation In Artificial Intelligence co-located with XVI International Conference of the Italian Association for Artificial Intelligence. Bari, Italy.

[2] Meter, C., Krauthoff, T., & Mauve, M. (2017, July). *discuss: Embedding Dialog-Based Discussions into Websites*. In International Conference on Learning and Collaboration Technologies (pp. 449-460). Springer, Cham.

[3] Krauthoff, T., Meter, C., Betz, G., Baurmann, M. & Mauve, M. (2018, September). *Dialog-Based Online Argumentation*. Computational Models of Argument, COMMA.

[4] Bex, F., Lawrence, J., & Reed, C. (2014, September). *Generalising argument dialogue with the Dialogue Game Execution Platform*. In COMMA (pp. 141-152).

[5] Bex, F., Snaith, M., Lawrence, J., & Reed, C. (2014). *Argublogging: An application for the argument web.* Web Semantics: Science, Services and Agents on the World Wide Web, 25, 9-15.

[6] Bex, F., Lawrence, J., Snaith, M., & Reed, C. (2013). *Implementing the argument web*. Communications of the ACM, 56(10), 66-73.

[7] Lawrence, J., Bex, F., Reed, C., & Snaith, M. (2012). *AIFdb: Infrastructure for the Argument Web.* In COMMA (pp. 515-516).

[8] Lawrence, J., Bex, F., & Reed, C. (2012). *Dialogues on the Argument Web: Mixed Initiative Argumentation with Arvina.* In COMMA (pp. 513-514).

[9] Rahwan, I., & Reed, C. (2009). *The argument interchange format.* In Argumentation in artificial intelligence (pp. 383-402). Springer, Boston, MA.

[10] Rahwan, I. (2008). *Mass argumentation and the semantic web.* Web Semantics: Science, Services and Agents on the World Wide Web, 6(1), 29-37.

[11] Rahwan, I., Zablith, F., & Reed, C. (2007). *Laying the foundations for a world wide argument web*. Artificial intelligence, 171(10-15), 897-921.

[12] Rowe, G., Macagno, F., Reed, C., & Walton, D. (2006). *Araucaria as a tool for diagramming arguments in teaching and studying philosophy*. Teaching Philosophy, 29(2), 111-124.

[13] Smith, S. P., & Harrison, M. D. (2002, April). *Improving hazard classification through the reuse of descriptive arguments*. In International Conference on Software Reuse (pp. 255-268). Springer, Berlin, Heidelberg.

[14] Kelly, T., & McDermid, J. (1998). *Safety case patterns-reusing successful arguments.*

Chapter 6

Various Efforts of Enhancing Real World Online Discussions

This chapter summarizes the contributions of the paper [37]:

Alexander Schneider and Christian Meter.
Various Efforts of Enhancing Real World Online Discussions
In Proceedings of European Conference of Argumentation (ECA),
Groningen, Netherlands, June 2019.
Acceptance ratio: 75%.

Tools and software developed for the purpose of conducting online discussions are often of academic nature and not aimed at real-world users. We on the other hand, developed software with the main aim to be used by real users on the Internet. After conducting several field experiments and experience gained through daily use of the software we feel confident that some insights on what does and does not work can be shared. The paper is our approach at sharing the gained knowledge for the purpose of improving discussion software that is often being developed in academic contexts.

6.1 Paper Summary

In the paper we reason that in order to improve online-argumentation, it needs to be usable and accepted by real-world layman users. To this end we introduce a tool-chain based on dialog-based argumentation. Dialog-based argumentation is the concept that a user is conducting a time-shifted dialog with a multitude of users at once. Every argument entered by a user is used to confront future users of the system to further a dialog-like structure. Exemplary for this concept, we discuss D-BAS and the experience gained from its field-experiments.

Afterwards, the concept of networked and reusable arguments is introduced. Networked arguments are shared between different providers of arguments which leads to the creation of a meta-graph between multiple argument networks. EDEN is introduced as an implementation of this concept and experiences with EDEN are shared.

Furthermore, we discuss alternative interfaces for dialog-based argumentation. To integrate dialog-based argumentation into arbitrary websites, a tool named *discuss* is presented. It makes it possible to use D-BAS as a back-end while having a lightweight integration in form of a script into any website, allowing users to conduct their argumentation anywhere. Lastly, *Jebediah* is presented as an interface for social networks with the same goals as *discuss*, but other means of integration.

6.2 Contribution

The main contribution of this paper is a push towards user-centered argumentation applications. Large parts of the community work on argumentation from an academic point of view and do not build systems aimed at real world users. Argumentation conducted by trained scholars is vastly different from argumentation used by an arbitrary person that has no formal argumentation training and likely not even any academic background. In this paper it is shown that there are many ways to create applications aiming at untrained users, which is an effort worth making.

6.3 Personal Contribution

Since this paper delivered an overview and experience report about technology that was already designed and developed before, the main work of this paper consisted in structuring and expressing the experiences of and interplay between the tools in writing. Alexander Schneider wrote the section regarding decentralized discussion networks as well as the Introduction and conclusion. Christian Meter wrote the sections concerning discuss and dialog-based argumentation as well as the related work. The section discussing auxiliary approaches was written in equal parts by both authors.

6.4 Importance and Impact on the Thesis

A great deal of the software and central thoughts that were developed during the creation of this dissertation is represented in this paper. It thus concisely sums up the software created as a product of our research.

Various Efforts of Enhancing Real World Online Discussions

Alexander SCHNEIDER and Christian METER [1]
Computer Networks Department
Heinrich-Heine-University Düsseldorf, Germany
firstname.lastname@hhu.de

Abstract. In this work we present a suite of software which enables gathering of natural language arguments from non-expert users of argumentation software without the use of NLP or other argument mining techniques. This is achieved by presenting the user with interfaces that prompt them to enter the data in a way in which it can be correctly added to an argument graph.

1. Introduction

In this work we present various efforts that try to answer the question of how to gather structured argumentation graphs from natural language discussions of non-expert users.

Gathering arguments through argument mining from natural language is an ongoing research effort that made a lot of progress in the last years. Despite this, considerable challenges need to be solved before argument mining is at its peak. Because of this we present different ways of gathering argument data from natural language discussions.

We tackle the problem by designing interfaces and systems which allow the user to input arguments, while the data is automatically structured into an argument graph in the background. We made several efforts to design dialog-systems which make use of this approach to interact with everyday users that are not argumentation-experts in any way.

A typical user is presented with an argument and the request to react to that argument (see Fig. 1). Participating users can then position themselves to that argument using statements introduced by other participants, thus strengthening the existing graph-structure or enter their own opinion. In that case the interface prompts them to input their argument in such a way that structured argument data is produced without further processing. This can be done by adding the new statement in the proper place in the argumentation graph, which the system can deduce from the selected choices of the user.

In this paper we describe three such interfaces, namely *Dialog-Based Argumentation System* (D-BAS) [1], discuss [2], and Jebediah [3]. The interfaces differ in their approaches. While D-BAS is a dedicated webservice for discussions which the user needs to visit, discuss allows the embedding of the interface into arbitrary websites. Jebediah enhances user experience by providing an agent for social networks with support for natural language processing. All these approaches share the same argumentation engine in

[1] Both authors contributed in equal parts to this work.

their backend, which is accessible via D-BAS' *application programming interface* (API) in the reference implementations.

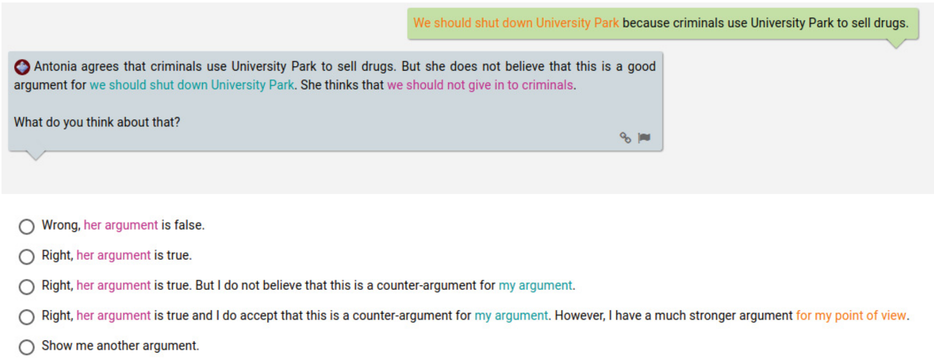

Figure 1. Gathering feedback during a confrontation in D-BAS.

The structured data created by the interfaces lends itself to reuse, and as a consequence we also present *Extensible Discussion Entity Network* (EDEN) [4]. EDEN is a reference implementation, which be used by discussion-providers to perform an automatic exchange of argumentation data. Examples of exchanged data are statements and arguments from the users, which can then be re-used in further discussions. We show that (automatic) reuse of argument data is possible and valuable.

As a last step in our pipeline we also provide a tool called *dabasco* [5], which enables the transformation of the gathered data into instances of *Argumentation Framework* (AF) [6], *Abstract Dialectical Framework* (ADF) [7] and ASPIC+ [8].

Thus, we present a complete pipeline of software projects which aid in the creation of natural language online discussions for non-expert internet-users, resulting in structured argumentation graphs that can be further used for analysis and other relevant processes. We reason that the pipeline presented in this paper is viable in conducting large-scale online discussions.

The rest of the paper is structured as follows: Section 2 describes dialog-based argumentation in general and D-BAS in particular. Following, section 3 introduces the reuse of arguments and an implementation for networking several dialog-based argumentation systems. In section 4 an alternative interface for integration of dialog-based argumentation systems into arbitrary web content is discussed. A social agent based interface and miscellaneous ways of exporting the collected data into other discussion frameworks are presented in section 5. In closing, we discuss related work in section 6 and end with our conclusions and future work in section 7.

2. Dialog-Based Argumentation

A lot of research in the argumentation community focuses on argument mining from natural language texts. Most argument mining research is done with the goal of creating a machine understandable corpus of arguments, which can be processed and used by algorithms. With that same goal in mind, we want to present a different approach. Instead

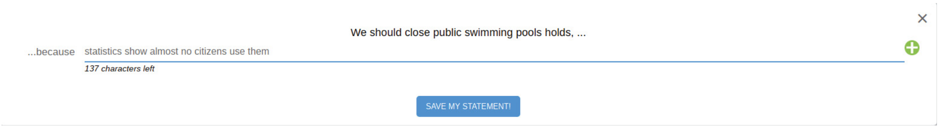

Figure 2. The view that is shown to users that are willing to enter their own counter-argument.

of letting human users debate with free text, e.g. in forums, and trying to mine the arguments after the fact, we want to engage them in a dialog-like exchange. This exchange still lets the users use natural language, but presents them with certain prompts at the same time. This compels the user to enter their thoughts in a structured manner, yielding arguments which can be added to an argumentation graph instantly.

2.1. The Idea Behind Dialog-Based Argumentation

Dialog-based argumentation was introduced in detail by Krauthoff et al. [9] and is best described as a multi-user dialog with a single system. Each user is confronted with an argument for some topic, that was not generated by the system but was entered by other users. Therefore, the user is basically engaged in a time-shifted dialog with other users. The main difference to “traditional” online discussions like forums is that the user is at all times being presented with a single argument, instead of e.g. a list. After the user reacts to the presented argument, a next argument made by other participants is chosen based on the user’s reaction. The reaction is then stored to be used in future interactions with the system.

Lets take a look at an example: The system contains a discussion with the topic “We should renovate the city’s library”. Now the system could present the interested user with several options, which confront the user with arguments *in favor of* renovating the city’s library or with arguments *against* renovating the library because, for example, it costs too much money. The user in turn can react to those arguments by either choosing counter- and supporting arguments that other users already made and the user feels are compelling, or by entering their own thoughts. This step is the crucial one which prompts the user to enter their argument in a structured manner as presented in Figure 2. Since the user is guided through a specially crafted menu, the system knows whether to input the user’s statement as an attack or support on a certain other statement, or if it is e.g. an undercut for some argument.

2.2. User-Focused Measures

The type of argument gathering, that we present with dialog-based discussion, relies heavily on the correct use of the system by the users. This leads us to focus on interface measures, which help the participants to navigate the system without issues.

Lets say a user is interested in the topic of whether to buy a dog or a cat. After the user expresses their interest in the topic, the system asks the user about what they want to debate in detail. Those options are for example “We should get a dog”, “We should get a cat” or “We should get another pet”. When the user selects the position they are interested in, they are prompted to state whether they are in favor or opposed to that option (or have no opinion, but want to see some arguments for that option). This is done, so the system knows whether the user interactions to come should be tallied as attacks or supports of certain arguments. Furthermore it enables the system to confront the user with fitting arguments from its database.

Anytime the user formulates their own arguments instead of reusing others, the system scans for similar arguments already made and presents them to the user. They can then choose to use one of the already present arguments to keep duplicates to a minimum. The dialog continues until the user does not want to have a discussion anymore, or until they reach a point in the discussion graph where there are no more attacking or supporting arguments left.

Duplicate, malicious or grammatically unsound arguments still make it into the system, since its main input source are typical humans. Those arguments can be moderated to make the experience a pleasant and engaging one for the users. Instead of using traditional moderators, the system implements the power of the masses. This has been included in D-BAS as a *decentralized moderation system* [1]. Users can e.g. mark duplicates or arguments violating the community's policies. Experienced users can then visit special randomized moderation queues, where they are presented with some of the marked arguments and can democratically vote whether to take action against those. Possible actions are for example "delete argument", "reformat argument" or "merge duplicates". If enough votes are tallied for a single option, it is executed.

2.3. Field Experiences

The dialog-based argumentation system D-BAS is online and free to use[2]. Besides experiences gathered from running the service, there also have been lessons learned from a formal evaluation through a field-study [10]. The study took place over 19 days and had 318 unique participants that visited the corresponding website. In this study the topic was how the computer science faculty could improve the bachelor's courses despite student numbers growing rapidly. All computer science students were invited to participate and the faculty promised to use the results as a base for future decisions.

During the experiment, more than 250 arguments have been created, which seems to suggest that users untrained in argumentation techniques are able to create a complex argument graph with the help of dialog-based argumentation. Parts of the resulting graph can be seen in Figure 3 and the associated data can be obtained online[3].

2.4. Application Programming Interfaces

D-BAS has two fully documented[4] and usable API options built-in to export the contents of a discussion and to allow third party applications to access the *Dialogue Game Execution Platform* (DGEP) parts.

The first endpoint provides authentication, authorization and the execution of discrete steps in the discussion. Applications can send requests to this endpoint to tell D-BAS about their current status of the discussion which then produces a response containing the next options and possible next discussion actions. Also sample text-responses are returned, which can then be used.

---

[2] `https://dbas.cs.uni-duesseldorf.de/`

[3] `https://dbas.cs.uni-duesseldorf.de/fieldexperiment`

[4] `https://dbas.cs.uni-duesseldorf.de/docs`

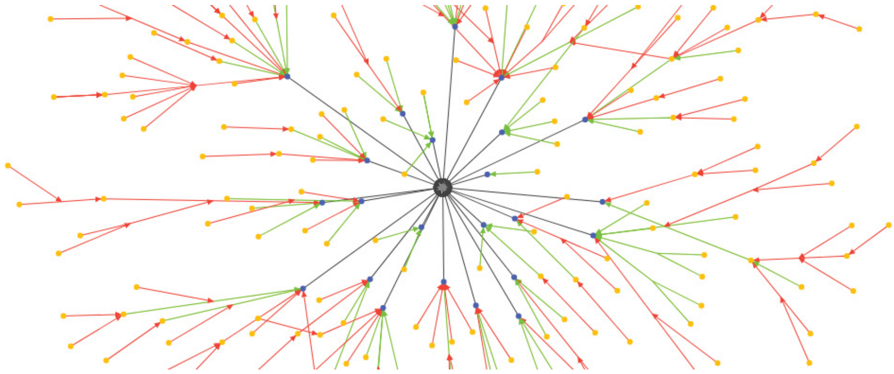

Figure 3. The graph resulting from discussions through D-BAS, discuss and Jebediah. Depicted is an instance from a real-world discussion. Colors: ● issue, ● positions, ● statements, ▬ supports, ▬ attacks

Data retrieval from our databases can be achieved using the second endpoint, which provides a GraphQL [11] API. This way people interested in the data can write their own queries to our databases to retrieve the public information from the hosted discussions.

3. Networked Arguments as a Resource

Through the use of dialog-based argumentation, people are able to create a wealth of arguments by following a dialog. But there are also scenarios where D-BAS has disadvantages. If we assume that, for example, several media outlets use dialog-based argumentation instead of simple list-like comments under their publications, each of them could run their own instances of dialog-based argumentation software. Now every user that wants to debate the same or a similar topic at different media outlets, is confronted with repeating arguments they are already familiar with. This would almost certainly happen due to the nature of how dialog-based argumentation is conducted. Furthermore, arguments made at one instance will never be seen on another, no matter how insightful or well worked out they may be. This section presents our thoughts on how to tackle these and related challenges.

3.1. Distributing and Versioning Arguments

We call every host, from the before-mentioned scenario, running their own dialog-based argumentation software, an *aggregator*. To put it in another way: an aggregator is an entity which provides content and the space to discuss it. To allow distribution of arguments, every aggregator can join a distribution network. Aggregators may have differing policies about which arguments are valid according to some rules or community standards. Hence, flooding the arguments to all aggregators in the network is unwise, because not all instances have the same policies and would be willing to receive certain arguments. Moreover, aggregators possibly want to keep the intellectual rights on arguments devised on their platform. Thus every argument needs to reference which aggregator is the authoritative instance for it. This means, that the arguments stay property of the differing aggregators, but still can comprise a single argumentation graph spanning over different physical and logical entities participating in the argument network. To allow other participants to propose changes to arguments, that they are not authoritative of, we need to introduce versioning. As presented by Meter, Schneider and Mauve [4] one can use a decentralized version-tree which is already known for versioning source-code. This means, that every argument has a pointer to its predecessor if one exists. Any changes can be proposed at once without violating or changing the original argument by creating a changed version which points to the original as its predecessor. The authoritative aggregator can decide whether to accept any of the proposed updates and incorporate them into the official version. But even in that case, there will be a new version from the authoritative source, since all arguments are created immutable.

3.2. EDEN: Extensible Discussion Entity Network

An exemplary implementation of a distributed argumentation network powered by aggregators is EDEN which was presented in detail in [4]. EDEN was developed in Clojure, a functional language on the JVM. Furthermore, we pursued a modular approach with EDENs architecture, which splits it up into four distinct modules – interface, discussion platform, database and aggregator core – which can be interchanged as long as the new module adheres to the proposed interfaces between the major parts.

The interface is tasked with guiding the user through the dialog-based argumentation. A database stores and persists the locally needed arguments. It can also provide features like semantic search on the arguments. The discussion platform is the piece of software that provides the internal logic on how to conduct the dialog-based argumentation, also known as DGEP. In the default case EDEN utilizes D-BAS as a DGEP. An aggregator core coordinates the flow of arguments between the different modules as well as between aggregators.

Communication between aggregators is handled in two parts. First, there is a REST API, which provides aggregators with the ability to actively query for discussion entities like arguments and their interrelations. As a second option a publish/subscribe queue exists, which automatically updates entities from known aggregators. For example if aggregator B requests some argument X on the topic of dogs from aggregator A, they also subscribe to the corresponding queues. When an update for X is available, B automatically gets informed about the update by A via the queue. Different update forms can be used. Instead of updates on queried arguments, B could receive notifications every time there is a new argument on the topic of dogs, to broaden its repertoire.

4. discuss: Embedding Dialog-Based Argumentation into Web-Contexts

One of the first applications using the API of D-BAS, is *discuss* [2]. discuss provides a minimal discussion interface to interact in the same flow as we have seen it in D-BAS, with the distinction, that it can be embedded in every web-context utilizing a JavaScript environment. This is intended to be used, for example, in online newspaper articles, which ask the readers to start a discussion in the comment sections. But since comment sections do not provide any structure, this approach could bring a significant improvement, because of the structural manner how the arguments of the users are being gathered.

Without having to leave the current scope, discuss provides (1) direct interaction with the author's arguments, (2) jumping into the discussions, where other participants

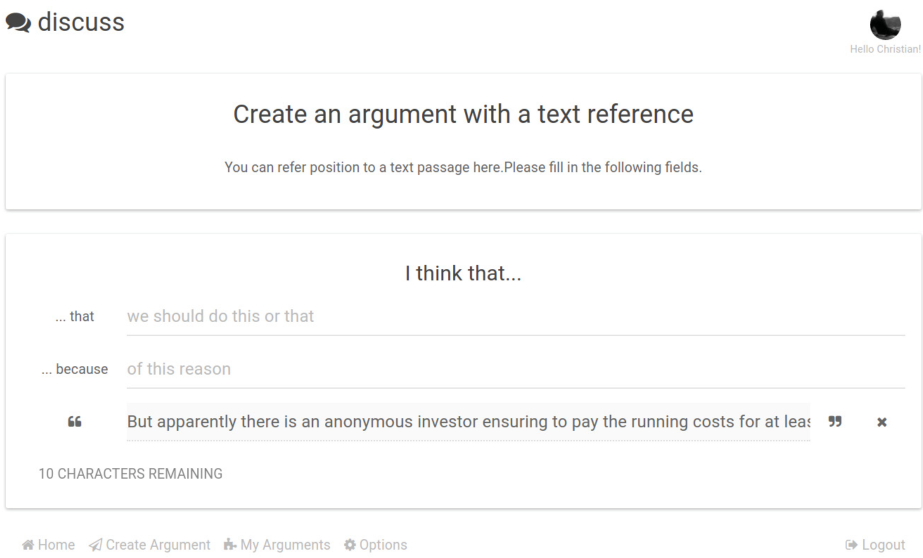

Figure 4. discuss: Create a new argument with a reference to a passage in the author's article.

interacted with the article, (3) enabling discussions in our proposed dialog-based flow (see 2.1) and (4) connect to the EDEN network.

4.1. *Interacting with the Author's Arguments*

One of the core functions of discuss is to directly interact with the author's article. Selecting an interesting part of a text passage opens up a dialog, where the reader can create a new argument with the selected text as a *reference* (see Fig. 4). Internally, the creation of an argument in this way is the same procedure as adding a new position in D-BAS, which introduces a sub discussion in the context of the discussion topic.

4.2. *Jumping into the Discussion*

Interactions with the article, which created a new argument with a reference to parts of the article, are highlighted so that the user sees an interactive element on the website (see Fig. 5). These references provide an entrypoint to the discussion, where the user's argument has been used. Also other arguments, which referenced the same text passages, are listed and users can decide where they want to jump into the discussion.

4.3. *Dialog-Based Discussion Flow*

We omit the selection of the initial positions in discuss, because we encourage to directly jump into the discussion via a reference in the text, i.e. hook into a pre-existing argument from a user, or by selecting a text-passage, i.e. create a new argument referring to the text. After the initial step, discuss presents the classical discussion flow which we have already seen in D-BAS (see 2.1). Specifically, this means that we conduct a dialog with the users and present those arguments, which have been posted about the argument from the article.

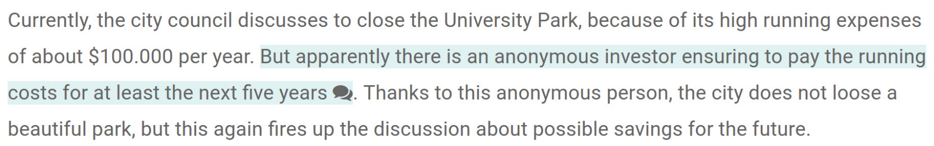
Currently, the city council discusses to close the University Park, because of its high running expenses of about $100.000 per year. But apparently there is an anonymous investor ensuring to pay the running costs for at least the next five years. Thanks to this anonymous person, the city does not loose a beautiful park, but this again fires up the discussion about possible savings for the future.

Figure 5. Text passage from an article, which has been used in an argument. A click on it opens the interface to jump into the discussion.

4.4. EDEN Integration

Besides the described functions, discuss can be used to connect to the EDEN network (see 3.2). D-BAS is then solely used as an DGEP for the steps in the discussions, whereas the arguments are being fetched from EDEN. This mechanism allows to retrieve and collect arguments from different locations and discussions, which can then be used in the current article's discussion.

5. Experiences With Auxiliary Approaches

Based on the presented tools, we felt the need for auxiliary applications. One is *Jebediah*, an alternative interface into dialog-based online discussions enabling users to discuss matters through chatbots and voice assistants. Furthermore we present *dabasco*, which allows the data generated trough D-BAS and its applications to be converted to other discussion frameworks for further use.

5.1. Jebediah

A vast part of online discussions takes place on social media platforms. Jebediah [3] is an interface which enables users of those platforms to take part in dialog-based online argumentation through chat-bots and voice assistants. Classifying the user's input is realized with the help of Google's Dialogflow platform [12], which is an Artificial Intelligence processor that tries to match the natural language input against predefined and pre-trained rules. The matching-process has the goal to produce structured data and the resulting data is being sent to a dialog-based argumentation software, like D-BAS. It returns a response, which is then again formatted and forwarded to the user through the chat-bot (see Fig. 6). This is still a highly experimental feature, which works most of the time but certainly can be further improved upon. Nonetheless, it would be interesting future work to test how users feel when discussing topics with a bot instead of a text-interface.

5.2. dabasco

The last step in our pipeline is the export of the generated data. Exports are useful to utilize collected argument data for further analysis. Building on the fact that some established tools by the community expect certain formats, Neugebauer developed an export interface called *dabasco* [5]. This way it is possible to export AF, ADF, and ASPIC+ data which was converted from D-BAS' data structure. dabasco uses D-BAS' API and provides the first 3rd party application interacting with our software stack.

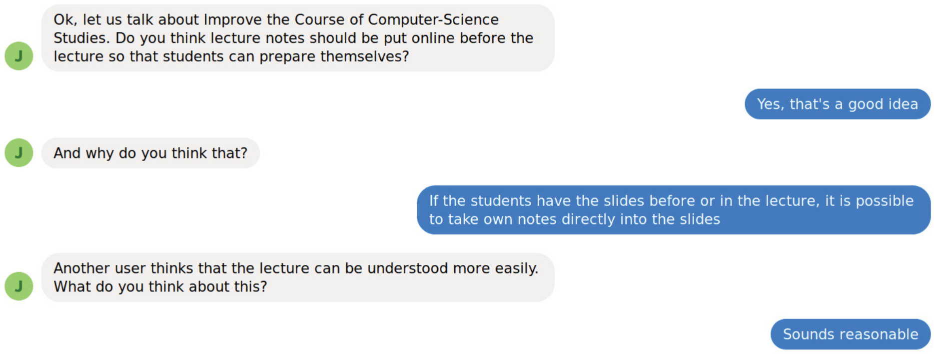

Figure 6. Left side: Dynamically produced text messages from Jebediah, right side the user's answers in the Facebook Messenger.

6. Related Work

Tools for facilitating online argumentation have been described and developed before. The set of tools that is most like the proposed pipeline is the argument web [13]. We build on similar ideas of a unified structured web of arguments and are not striving to compete with the argument web but to be compatible to magnify the extend of the argument network. AIFdb, developed by Lawrence et al. [14], is in spirit akin to EDEN regarding collecting arguments from differing sources, but differs in aspects of centralization and the kind of arguments collected. Other approaches at structuring arguments, include Carneades [15], Deliberatorium [16] or OVA as introduced by Snaith et al. [17]. The difference to is that none of those are based on dialog-like argumentation. Most of these tools focus on the whole discussion, whereas our smallest entity is the statement, which could be put together to an argument and the put into context, e.g. of a discussion.

7. Conclusion

In this paper we presented a complete pipeline for gathering, sharing and exporting user-generated arguments. We introduced D-BAS, a system that conducts discussions by simulating a dialog with other users. A field-study verified that this approach yields a structured argumentation graph and even untrained users were able to use our software in a productive way. Moreover, we presented discuss, which enables arbitrary websites to integrate a D-BAS-style discussion and Jebediah, which does the same for artificial assistants. To share the generated arguments between instances of D-BAS, we use EDEN, which provides the ability to decentralize an argumentation network. Lastly, dabasco allows the export of D-BAS arguments to different argumentation frameworks, which can be used for further calculations.

This paper showed that a pipeline for gathering structured argumentation from natural language without argument mining is possible and how such a pipeline may be structured.

For future work we plan to conduct field experiments that make use of the complete pipeline to test its efficiency. We furthermore are developing tools that harness the dialog-based stack to conduct discussions with the goal of finding and voting on solutions for e.g. the budgetary allocation of a city.

References

[1] T. Krauthoff, C. Meter, G. Betz, M. Baurmann, and M. Mauve, “D-BAS – A Dialog-Based Online Argumentation System,” in *Computational Models of Argument*, September 2018, pp. 325–336.

[2] C. Meter, T. Krauthoff, and M. Mauve, “discuss: Embedding dialog-based Discussions into Websites,” in *Human Computer Interaction International: Proceedings of HCI*, 2017.

[3] C. Meter, B. Ebbinghaus, and M. Mauve, “Jebediah – Arguing With a Social Bot,” in *Computational Models of Argument*, September 2018, pp. 467–468.

[4] C. Meter, A. Schneider, and M. Mauve, “EDEN: Extensible Discussion Entity Network,” in *Computational Models of Argument*, September 2018, pp. 257–268.

[5] D. Neugebauer, “DABASCO: Generating AF, ADF, and ASPIC instances from Real-World Discussions,” in *Proceedings of the 2018 Conference on Computational Models of Argument*. IOS Press, 2018.

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

[7] G. Brewka and S. Woltran, “Abstract dialectical frameworks,” in *Twelfth International Conference on the Principles of Knowledge Representation and Reasoning*, 2010.

[8] M. Caminada and L. Amgoud, “On the evaluation of argumentation formalisms,” *Artificial Intelligence*, vol. 171, no. 5, pp. 286–310, 2007.

[9] T. Krauthoff, M. Baurmann, G. Betz, and M. Mauve, “Dialog-based online argumentation.” in *COMMA*, 2016, pp. 33–40.

[10] T. Krauthoff, C. Meter, and M. Mauve, “Dialog-Based Online Argumentation: Findings from a Field Experiment,” in *Proceedings of the 1st Workshop on Advances in Argumentation in Artificial Intelligence*, vol. 2012. CEUR-WS, 2017, pp. 85–99.

[11] “A query language for apis.” [Online]. Available: https://graphql.org/learn/

[12] “Dialogflow.” [Online]. Available: https://dialogflow.com/

[13] I. Rahwan, F. Zablith, and C. Reed, “Laying the foundations for a world wide argument web,” *Artificial intelligence*, vol. 171, no. 10-15, pp. 897–921, 2007.

[14] J. Lawrence, F. Bex, C. Reed, and M. Snaith, “Aifdb: Infrastructure for the argument web.” in *COMMA*, 2012, pp. 515–516.

[15] T. F. Gordon and D. Walton, “The carneades argumentation framework–using presumptions and exceptions to model critical questions,” in *6th computational models of natural argument workshop (CMNA), European conference on artificial intelligence (ECAI), Italy*, vol. 6, 2006, pp. 5–13.

[16] M. Klein and L. Iandoli, “Supporting collaborative deliberation using a large-scale argumentation system: the mit collaboratorium,” 2008.

[17] M. Snaith, J. Devereux, J. Lawrence, and C. Reed, “Pipelining argumentation technologies.” in *COMMA*, 2010, pp. 447–453.

Chapter 7

discuss vs. Disqus: Evaluating Dialog-Based Discussions Against a Comment-Based System

This chapter summarizes the contributions of the paper [38]:

Christian Meter, Alexander Schneider, Marc Feger, Jan Steimann, Martin Mauve:
discuss vs. Disqus: Evaluating Dialog-Based Discussions Against a Comment-Based System
Submitted to Computational Models of Argument (COMMA 2020),
Perugia, Italy, September 2020.
Acceptance ratio: under review during publication

Many dialog-based systems have been built over the last years. Starting from the stand-alone D-BAS, to discuss, which can be used as an easy to integrate frontend, to a whole ecosystem of applications, which include EDEN. Although, Krauthoff et al. [19] already tested how users react to a basic stand-alone implementation of dialog-based discussions, there still was no comprehensive study directly comparing dialog-based approaches to other forms of discussions on the Internet.

With this paper, we aimed to close this gap. Gathering student participants, we conducted a comparison of *disqus*, which is a comment-based solution, with *discuss* and EDEN as a front- and backend, respectively, for the dialog-based approach. We gathered the data by letting the participants fill out questionnaires after they have had time to discuss three articles using the provided tool.

The results were mixed, but promising and show that a dialog-based approach can hold its own against a comment-based system.

7.1 Paper Summary

In this paper we tried to evaluate a current discussion-based approach to Internet-debate with the most prevalent comment-based one. The key contributions of this paper are:

1. A working setup for comparing different forms of internet discussions, integrated into news articles.

2. Definition of hypotheses which can be used to evaluate real-world discussion systems.

3. The first direct comparison between dialog-based and comment-based discussion systems and a presentation of its results.

Firstly, the paper motivates the need of a proper comparison of the two systems. It needs to have a realistic baseline which can be pursued in the improvement of dialog-based systems. It is concluded that the best way to compare the systems effectively is under a controlled environment like a lab setting with predefined hypotheses. The hypotheses will be evaluated based on a questionnaire that is answered by the study-participants and by analyzing the content produced.

Following the motivation, the experiment itself is introduced. We used the two systems *discuss* and *disqus*. The participants who are in one of two possible types of group see the exact same three articles, which center around vegetarian diet and the environment. Only the system which is embedded to facilitate discussions differs between the groups. Before we conducted the experiment, we created eight different hypotheses. Those center mainly around the usability of *discuss* compared with a well known approach which was presented in the form of *disqus*. Besides usability, the hypotheses also centered around whether dialog-based systems foster more objective, fact-based or respectful discussions. Participants of the study were recruited on campus, and they were all students. They were paid €10 for their 40 minutes of participation. No moderation of any kind was conducted.

Overall, 62 students participated in the experiment in 11 groups. Out of all students, 27 were assigned to the *disqus*-group which was used as a control, while the remaining 35 were assigned to the *discuss*-group. We compared the differences in questionnaire-results between group types. To test for statistical significant differences, the Mann-Whitney-U-Test was used. Most hypotheses, which aimed to show that *discuss* provides a user experience comparable to comment-style systems, did not hold. On the other hand, the participants in the *discuss*-group felt that the people discussing the articles were more respectful with each other. They also produced more than twice as many arguments per user, compared to the control-group, which used *disqus*.

The data strongly implicates that there is need for further improvement of the user experience (UX) of our dialog-based systems. When we further analyzed the produced arguments by annotation through 4 annotators, we noticed that the resulting number of arguments produced by the comment-based system had a significantly higher variance between the annotators. In other words: There was a lot more disagreement between annotators which statements contained arguments, and how many statements they contained, compared to the statements produced by *discuss*.

7.2 Contribution

The first contribution of this paper is a setup for a controlled experiment to compare two types of discussion systems. We describe how we minimize the differences so that the only alteration between test and control group is the tool used. We also provide hypotheses which can be used to especially evaluate dialog-based systems.

Furthermore, we provide all raw data gained from the study and a discussion of the results. We can demonstrate that dialog-based discussions enable every-day users to produce a high number of arguments during a discussion, which has not been shown before.

7.3 Personal Contribution

The author of this thesis, Alexander Schneider, and Christian Meter designed and conducted the experiment in equal parts. They also wrote all parts of the resulting paper, except for the “Lessons from Annotator Differences” subsection, whereby Christian focused on the hypotheses and Alexander on the analysis of the results. Marc Feger and Jan Steimann helped recruit participants for the experiment, contributed the “Lessons from Annotator Differences” subsection, and assisted with the analysis of the results.

Martin Mauve contributed during the experiment design and gave extensive feedback regarding the editorial content.

7.4 Importance and Impact on the Thesis

With this paper we can conclude the work we did in the previous four papers. It neatly ties everything together by demonstrating the viability of the dialog-based ecosystem we built with EDEN, D-BAS and *discuss*. Equally important, the results we gained with this paper also show us various pathways for future work, after the completion of the doctoral thesis.

We set out to build a system which is capable of treating arguments as reusable objects in a network, while simultaneously facilitating factual, objective discussions for the everyday user. This paper confirms that we reached this goal.

7.5 Note on Following Paper

The following paper is a manuscript that was submitted to the Proceedings of Computational Models of Argument (COMMA) 2020. This paper has at the point of publishing of this dissertation not been published anywhere else and is awaiting peer-review.

September 2020

discuss vs. Disqus: Evaluating Dialog-Based Discussions Against a Comment-Based System

Christian METER [1], Alexander SCHNEIDER[1], Jan STEIMANN, Marc FEGER and Martin MAUVE

Computer Networks Department
Heinrich-Heine-University Düsseldorf, Germany
firstname.lastname@hhu.de

Abstract. In this work we present the results of a hypotheses-guided lab experiment comparing the discussion of online newspaper articles by means of regular comment sections versus the use of our own, dialog-based approach. We show that the main problem of our approach is usability and user interface design. At the same time we can prove that it has a large positive impact on the number and clarity of users' arguments. As a consequence we reason that more effort should be spent on user interface and user experience design of systems that support online argumentation.

Keywords. argumentation, argumentation system, online discussion, dialog-based, web-application, study

1. Introduction

The focus of research on online argumentation, so far, has mainly been on either theory or on designing novel systems. Some of those systems have then been put to the test by using them in lab or real-world settings. Typically, the authors of those systems report that the tests were quite successful. However, at the same time, the collective research in this area has had limited impact on how online discussions and argumentations are conducted in the real world. In the vast majority of real-world applications, some form of forum- or comment-based system is still used. An approach that our research community thinks of as being deeply flawed.

In an attempt to shed some light on why that might be the case, we have conducted a hypothesis-guided lab experiment. In this experiment we compared discuss [1], our own approach to support online argumentation, with Disqus[2], a commonly used comment system.

The main findings presented here are as follows. First, we provide very detailed information regarding the advantages and drawbacks of using discuss in comparison to

[1]Both authors contributed in equal parts to this work.
[2]`https://disqus.com`

Disqus. Second, we demonstrate that hypothesis-guided lab experiments provide important insights. Finally, the results of our experiment indicate that our approach has significant potential to outperform forum based systems. However, it is held back because participants are not familiar with our system and the user interface is not good enough to compensate for this.

The remainder of the paper is structured as follows: Section 2 describes the experiment setup, our research questions and hypotheses. In Section 3 we present the data gathered through the experiment and a statistical evaluation of the results. Following, Section 4 contains our interpretation of the results. Related work is discussed in Section 5. We conclude the paper with a summary and an outlook in Section 6.

2. Experiment

2.1. Argumentation Software

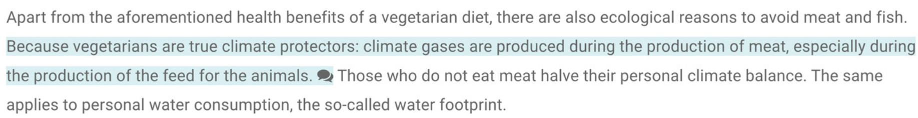
Apart from the aforementioned health benefits of a vegetarian diet, there are also ecological reasons to avoid meat and fish. Because vegetarians are true climate protectors: climate gases are produced during the production of meat, especially during the production of the feed for the animals. Those who do not eat meat halve their personal climate balance. The same applies to personal water consumption, the so-called water footprint.

Figure 1. A text reference created with discuss. Clicking on the highlighted part, jumps into the discussion shown in Figure 2.

We used two different software tools in our study. The first is discuss, our own tool for dialog-based online discussions. In discuss users can mark a section of a web page and attach an argument to it. As shown in Figure 1 this section becomes highlighted and other users can click on it to see the attached arguments and enter a dialog-based discussion.

In dialog-based discussions the user is shown an argument of another user and can react to it. This is depicted in Figure 2. One possible reaction is to attach another argument. Based on the reaction the user is then confronted with the next argument. In this way the user conducts a dialog with the system, while the system represents all users that have already added arguments in the past.

The other software is Disqus, a popular tool to embed hosted comment sections into websites. Users can add their comments and reply to others, see Figure 3. Disqus was used because it provides a similar feature-set to discuss, e.g. inter-article discussions, which makes it a good comparison.

2.2. Research Hypotheses

The main goal of our study was to get a good understanding whether users would accept or possibly even prefer discuss as a replacement for common commenting tools such as Disqus. To this end we set up the following series of hypotheses before conducting the experiment.

H1 *Using "discuss" is as intuitive as using "Disqus".*

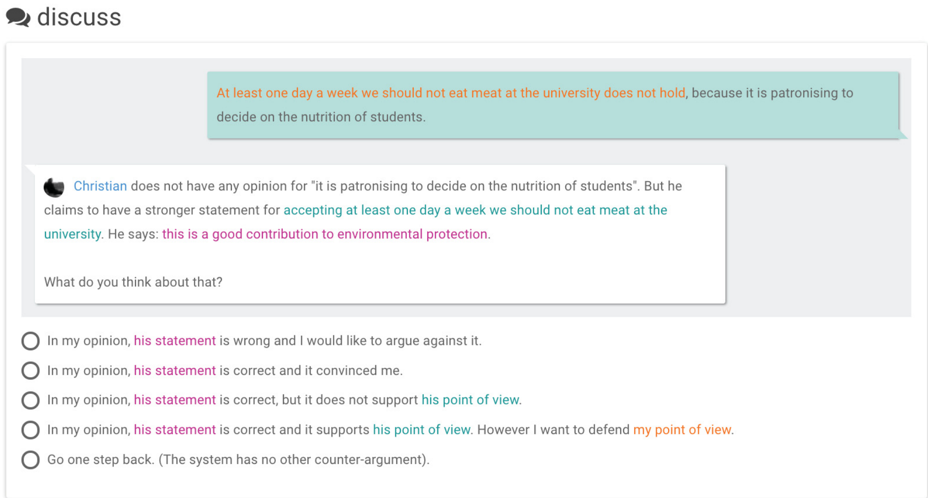

Figure 2. A reaction-step in a discussion with discuss.

H2 *"Discuss" offers the necessary flexibility to comment on a specific aspect of the article.*

H3 *Other users' contributions are of interest to the user.*

H4 *It is easy for users to get used to the "discuss" user interface.*

H5 *With "discuss" it is not easier to understand the context of an argument made by another participant.*

H6 *With "discuss" it is not easier to gain a good overview of a discussion.*

H7 *It is not helpful to use "discuss" to argue across articles.*

H8 *The ability to reuse arguments is used more frequently with "discuss".*

The reason why we used a mixture of positive and negative (in relation to discuss) hypotheses is that we generally tried to formulate the hypotheses in a way as to be able to disprove them in a statistically significant way and at the same time learn how to best proceed with the development of discuss.

H1 and H4 aim at measuring the subjective feeling of the users to compare discuss and Disqus regarding accessibility. This was important to us since we anticipated that our own tool might have problems in this area because we are no experts in user interface design.

Discuss allows users to directly interact with the text of the articles. We therefore expect it to do better in regard to commenting on one specific aspect of the article. This is captured by H2. We expect many interactions with passages in the article and therefore a good result when evaluating this hypothesis for discuss — at least better than Disqus.

H3 targets the general interest in the opinion of other users. Since the users are participating in the discussion voluntarily, we are expecting both groups to have a high interest in the topic and the opinions of other users.

By asking for the context of an argument, we are expecting in H5 that the argument's context is clearer in discuss than in regular comment sections. H6 is set up to prove a

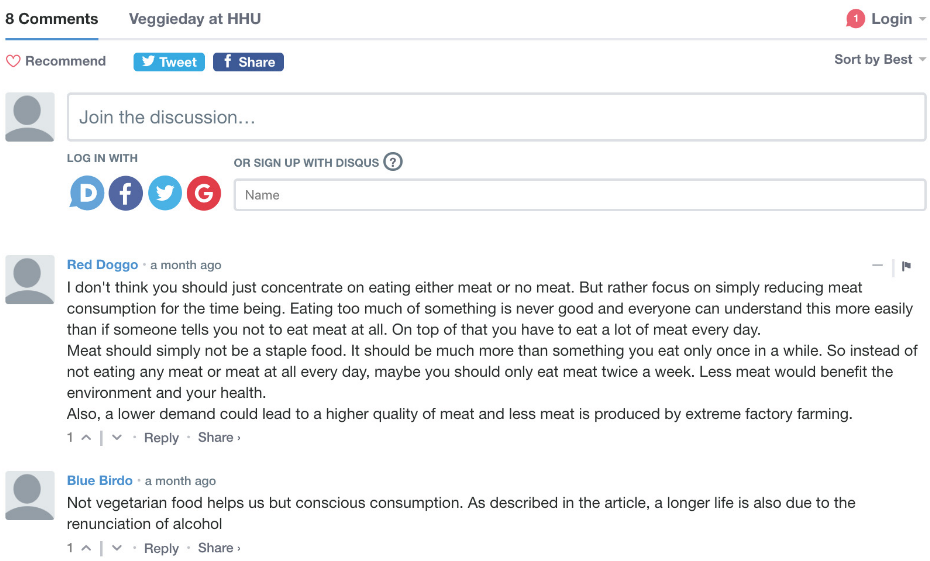

Figure 3. Two sample comments of the discussion with Disqus.

similar result, by targeting an overview of the complete discussion. We anticipated that discuss should perform better than Disqus in this area as well.

H7 refers to the mechanism for inter-article discussions. Both groups have the technical means to use this feature. We expect better results for discuss, because of a deeper software-integration of this feature.

Re-using arguments to reduce redundancy is one of the key goals in the dialog-based discussion approach. Thus, discuss has mechanics implemented to reuse previously submitted arguments, which are covered by H8. We expect at least some amount of arguments to be reused as the discussions develop.

2.3. *Experimental Procedure*

Students in groups from four to eight people were invited to join the discussion in person at one of our computer labs (see more details in Section 3.1). The computers were prepared to have a clean browser and three online articles about vegetarian diet opened. These articles either integrated discuss or Disqus. Participants using Disqus are from this point on referred to as the *control-group*. All browsers were opened so that users could directly participate in the discussion. The initial state of the discussion contained two arguments provided by us as a starting point.

Each participant had an own computer with random credentials for participation. It was not possible for them to look on the screen of other users, and they were instructed to only communicate online. A text-tutorial was attached to their screens to explain in a few words how to use the tools. The discussions were saved and reset to the initial state after each run.

Both tools allowed reading arguments of other users and to add new ones. The control-group was able to discuss in the typical comment-reply pattern as it is broadly used in online news media. Discuss users were guided through the discussion as it is known of *Dialog-Based Online Argumentation [2,3]* (DBOA).

Participants were allowed to browse the Internet freely during the experiment, e.g. to acquire background information or to look up facts. We did not moderate any of the arguments and did not participate in the discussion. Only technical support was given by us if something was unclear.

Each group of participants was first instructed about the procedure of the study with the exact same text read aloud by one of the authors. Afterwards, they had 30 minutes to discuss, ten minutes to answer the questionnaire and were in the end awarded with € 10.

We announced the study in several lectures, posted flyers on bulletin boards, posted on Twitter and came into direct contact with the students on our campus. All of them participated freely in the discussion. The participants could choose between eleven dates, all taking place within three weeks. A twelth date that we provided to even the number between the groups did not get any reservations.

3. Results

We obtained two kinds of results from the experiment. The first kind are subjective ratings from the participants, regarding their perceptions about the software they used and the discussion they led. For this the participants were presented with a questionnaire containing assertions, which they had to rate on a five-part *Likert Scale* [4], ranging from one, representing "I absolutely disagree" to five, representing "I absolutely agree". The questions and the results are shown in Table 1. We used the Mann-Whitney-U-Test to test for statistical significance of the differences in rating. We further reject or accept our hypotheses based on those results that are significant. A part of the questionnaire was only answered by users in the discuss-group, since the questions targeted properties of discuss specifically. The second kind of data is the data produced directly by the users, for example the number and content of arguments. We annotated the statements that the participants produced with four annotators and compared, e.g., the number of arguments per statement between the control- and the discuss-group.

#	Question	Average		Median		Variance		MMW
		discuss	control	discuss	control	discuss	control	
1	I was personally interested in the topic	4.229	4.148	4	4	0.462	0.8669	$p: 0.9876$
2	I would participate in a discussion for a similar topic	4.4	4.385	5	5	0.5257	0.7751	$p: 0.8253$
3	I understood how to participate in the discussion	4.086	4.692	4	5	0.8784	0.5207	$p: 0.0041$
4	It was easy to comment on a specific part of the article	3.667	4	4	4	1.434	0.963	$p: 0.3293$
5	The comments of other users interested me	4.235	4.296	5	5	0.7682	0.8752	$p: 0.6772$
6	I understood how the discussion worked	3.647	4.63	4	5	1.287	0.4554	$p: 0.0005$
7	I had the feeling that a lot of the comments did not fit the topic	2.429	1.538	2	1	1.445	0.7101	$p: 0.0019$
8	I had the feeling that I had a good overview of the discussion	2.771	3.481	3	4	1.319	0.7682	$p: 0.0154$
9	I think that multiple articles for the same topic enriched the discussion	3.909	4.346	4	5	1.355	0.8417	$p: 0.1507$
10	I learned something through the comments of other participants	3.086	2.808	3	2	1.678	1.386	$p: 0.3936$
11	I gained a new perspective regarding the topic through the discussion	2.657	2.519	3	2	1.425	1.805	$p: 0.5895$
12	I lost track of the content of the discussion	3.029	1.846	3	1	1.628	0.9763	$p: 0.0005$
13	The participants treated each other respectfully	4.781	4.444	5	5	0.2334	0.8395	$p: 0.1561$
14	Highlighting sentences inside the article was disruptive	1.857	-	2	-	1.094	-	-
15	The suggestion of arguments was helpful	2.853	-	3	-	1.831	-	-
16	The tool "discuss" helped the discussion	3.212	-	3	-	1.379	-	-
17	I understood how to navigate through the discussion	3.6	-	4	-	1.154	-	-
18	"discuss" enables better discussions than traditional comment boxes	3.125	-	3	-	1.234	-	-
19	I think that "discuss" leads to a more intense reflection of the arguments	3.871	-	4	-	1.209	-	-
20	"discuss" is too complicated and I got lost	2.5	-	3	-	1.132	-	-
21	I think "discuss" leads to a more respectful discussion between the participants	3.267	-	4	-	1.596	-	-

Table 1. Translations of the questions from the survey the participants had to fill out after the discussion. We used a 5-point Likert scale for each question, ranging from 1: "I absolutely disagree" to 5: "I absolutely agree".

In the following, we highlight the differences between the two groups, discuss whether those differences are statistically significant and in cases where they relate to our hypotheses, whether they can be used to confirm or reject them.

3.1. Participant Data and Composition

Overall, 62 students participated in the experiment in 11 groups. Participants were allocated to five control-groups with 27 members in total, while 35 students were presented with the discuss-software in six groups. The age distribution ranged between 17 and 50 with a mean of 23.17 for the control-group and 23.68 for the discuss-group. In the control-group 15 participants were female, 10 male, one person chose “not specified”, and one did not fill out the corresponding form. For the discuss-group, 18 participants were female, 16 male and one chose “not specified”. Control-group participants were from 13 different degree courses and had a semester average of 3.889 while the discuss-group participants were from 16 different degree courses and averaged 5.057 semesters.

3.2. Discussion Quality Perception

We used the first set of questions (1-7) from Table 1, which were answered by participants in both the control- and the discuss-group, to measure how they perceived the quality of the discussion itself. At first, Questions 1 and 2 asked the participants for their interest in the topic. Our results showed no difference between both groups, which implies that a predisposition to topic preference did not color the following results.

Question 3 (“I understood how to participate in the discussion”) was supposed to test for the intuitive usability of the software. Here discuss had a worse, statistically significant, outcome than the control-group software. This directly disproved H1, which was expected since the participants are used to comment-boxes and for the most part heard the first time about dialog-based argumentation during the experiment. Similarly, the results from Question 6 disprove H4 as well.

H2, and H3 on the other hand held, as denoted by the results from Questions 4 and 5. Incidentally, both of the supported hypotheses target inherent qualities of discuss. It was assumed that discuss would at least perform equally to conventional comment-boxes, which it did.

Participants that used discuss felt more strongly that the comments of others were unfitting, which is shown by Question 7, in turn implying that H5 does hold.

3.3. Overview of the Discussion

A second set of questions (8-13) tested if the participants were able to navigate the discussion or whether they could gain a rough idea what the discussion was about.

Two of the questions belonged together and should have a related outcome to gauge whether the participants where answering thoroughly or just clicked randomly. Those were Questions 8 and 12 as well as 10 and 11. In all cases the results did fit.

For Questions 8 and 12 the results for discuss were worse and therefore supported H6. Questions 10 and 11 seem to support this further, although the differences are smaller and not statistically significant. The ability to comment on several articles with the same software was queried by Question 9. Even though the results are slightly in favor of the control-software, they are not statistically significant and thus H7 is rejected.

Additionally, we presented the statement "*The participants treated each other respectfully*" which was not linked to a hypothesis and was included to gain a sense whether the participants felt respected. A difference of 0.34 in favor of discuss was not statistically significant ($p = 0.1562$).

3.4. Discuss-Specific Questions

The last set of questions (14-21) was only presented to participants in the discuss-group, as they reference certain features of discuss, which are not directly comparable to the comment software used with the control-group.

Results suggest that H8 does not hold, since the according survey question was on average answered with 2.853 points and a variance of 1.831. As a reminder: 1 represented that the participant felt no use at all for the argument suggestions while 5 represented that the participant felt the suggestions were very helpful. Other results in this section showed mediocre outcomes for features of discuss, except for a perceived heightened sense of critical thinking in regard to the arguments of the discussion in Question 19. Five of the participants in the discuss-groups acknowledged that they at least heard of the dialog-based argumentation style before. The other 30 participants were confronted for the first time with dialog-based discussions.

3.5. Content Difference

Besides the questionnaire answers, we also analyzed comments produced by the participants. Looking at the number of "statements", a user produced *4.88* on average for the control-group and *8.26* for the discuss-group. A "statement" is a typical comment, not regarding whether it contains an argument or not. It is important to state that in a lot of cases the discuss-software, by design, forces the user to enter two "statements". This is, for example, the case when the user adds a new argument and needs to provide at least one statement for the premise and one for the conclusion. 52.67% of the statements in the control-group were a direct reaction to the statement of another user, which suggests a high interactivity. The rate for the discuss-group is 100% and not comparable, because participants react to the statements of other users by design.

To gauge the total number of produced arguments, we used four annotators that worked through the statements and noted the number of arguments contained in them. A nonrestrictive definition was used to define an argument: It needed to contain at least one premise and a conclusion. We measured the inter-coder reliability through the Holsti method. The overall reliability was 76.29%, which is usually on the brink of acceptability for argument-annotation from natural text. When we look at the *Holsti Index* [5] for the statements from the control- and discuss-groups separately, we get a reliability of 55.47% and 88.96% respectively.

Depending on whether we take the lowest, highest or the average scores produced by the annotators, 247, 418 or 330 arguments have been produced in total across all groups, respectively. This means on average every participant produced 5.24 arguments. The spread of possible arguments (and thus the disagreement between the annotators) is significantly higher when only the control-groups are considered. Then we get 27 (lowest), 149 (highest) or 94 (average) arguments, with 3.48 arguments per participant. In contrast, the discuss-groups, which had 8 participants more, produced 220 (lowest), 269 (highest) or 236 (average) arguments, which results in 6.75 arguments per participant. Further analysis and explanation of this disparity is given in Section 4.

4. Discussion

The questionnaire data in itself produced mixed results regarding the previously stated hypotheses. Since not all have been rejected, we will now discuss conclusions that can be drawn.

4.1. Questionnaire Implications

One key result of the evaluation is that the hypotheses concerning the intuitive handling and usability of discuss (H1 and H4) were rejected and that it was not easier to understand the context of an argument (i.e., H5 was accepted). This result is important since it is very unlikely that a system with these problems will achieve widespread use in real-world environments. We believe that there are three main reasons for this outcome. First, participants are used to existing forum-based systems, therefore those systems have an implicit advantage regarding usability. Second, entering and interacting with arguments is likely harder than just writing and referring to plain text. Third, we are no experts in user experience and user interface design, thus it is very likely that both user experience and user interface are far from being optimal. Out of those three reasons only the last one can be changed. Improving the user interface and the user experience will therefore gain a very high priority in our future work.

Discuss got significantly worse grades for the statement "I had the feeling that a lot of the comments did not fit the topic", which supported H5. We expected scores to be better or at least equal to the control-group, since we assumed, that discuss enforces a more factual discussion with less off-topic comments. One possible explanation is that users are always confronted with a counter-argument to their last statement. Continuing the discussion this way, could produce a "rabbit-hole" effect, whereby the user is debating increasingly irrelevant seeming sub-issues. In contrast, using comment-boxes allows the user to see several comments at once and thus pick the more fitting ones. It was also easier in the control-group to keep the overview because of the lower number of arguments, which were all produced during the experiment. Therefore, the participants could keep track of all changes in the discussion, which is not always the case in bigger discussions.

An interesting observation is that users seem to perceive the participants of the discuss-version to be more respectful towards their peers. Although, this may only be a trend since the differences could not be proven to be statistically significant ($p = 0.1561$). This would be plausible, since discuss enforces a more strict argument-focused style of discussion, which causes the participants to use less *ad hominem* and other uncalled-for behavior. Users interact more on an argument-centric interface, instead of the typically personal message-based interface of comment-boxes.

Participants mostly agreed that discuss leads to a more intense reflection of arguments, which again makes sense since discuss focuses on arguments rather than personal opinion. Other interesting results were that most users did not feel disturbed by highlighting parts inside the article. This knowledge can be used in the future to strengthen objectivity in arguments by enabling the user to mark sentences as a kind of citation or direct reference. A similar observation has already been made by Mullick et al. [6]. They conducted experiments to test the hypothesis, that users mainly do not read the whole article and mostly comment on specific parts. Out of the 20 participants of this study 17 stated that they enjoyed the possibility to comment specific paragraphs instead of the whole article.

4.2. Produced Arguments

The most important result from the analysis of the content produced by the participants was the difference in the number of statements and arguments per participant. The participants in the discuss-group produce more than double the amount of arguments per user, which was to be expected as an outcome of an argument-focused interface.

Furthermore, it was also more clear-cut for the annotators what statements constituted one or more arguments. The *Inter-Annotator Agreement [7]* (IAA) between any two annotators was between 84.47% and 95.43% for statements from the discuss-group, while it was between 42.75% and 70.23% for the control-group. This hints at two things: Firstly, statements produced by discuss seem to contain more clearly structured arguments. Secondly, statements given by plain text commenting on an article seldom contain clearly identifiable arguments.

We believe that the main reason for this is that a lot of the comments produced by users in the control-group were written in a way, that did not state their intentions explicitly, but more or less implied what they wanted to express. This has the effect that it depends on the reader and their current state of mind whether they register the possible arguments contained in the comment. Another reason could be that when the users are not nudged towards producing arguments, they simply only state their opinions or any kind of off-topic comments without the intention of starting or participating in a conversation or debate. Related studies show this as well: When presented with traditional commenting options users often participate for the purpose of asking questions, provoking others, providing new perspectives [8] or for purely entertainment value [9].

4.3. Lessons from Annotator Differences

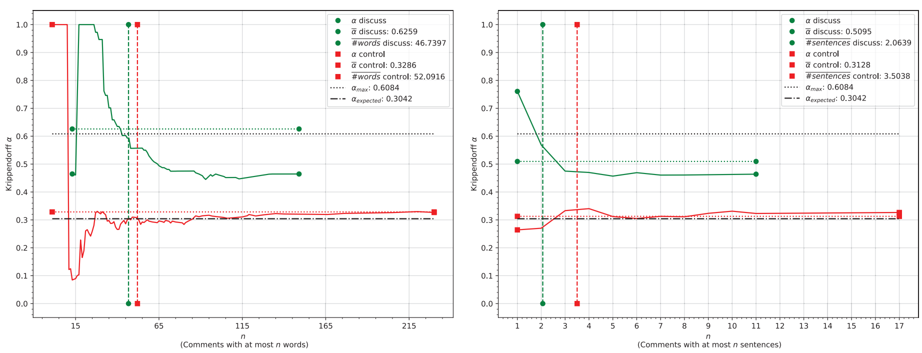

Figure 4. Development of Krippendorff α against the number of words and sentences in the comments.

Additionally, the systems can be compared using the argument annotations and the resulting IAA. For this purpose, we have annotated each of the 350 comments with the corresponding number of arguments. In order to not distort the annotation and to preserve the individual view of the annotators, no correction phase was carried out. As an additional IAA measure, we used the study-specific Krippendorff α in combination with the ratio-distance function [10,11]. Overall, a match of $\alpha_{global} \approx 0.535$ was achieved on the data of both groups. The highest result with $\alpha_{max} \approx 0.608$ was achieved by omitting an annotator. Accordingly, the lowest result was also achieved with $\alpha_{min} \approx 0.477$. In the control-group, an agreement of $\alpha_{control} \approx 0.327$ for all annotators with a total of 131 comments was observed. Analyzing the 219 comments obtained by discuss, $\alpha_{discuss} \approx 0.464$ could be measured. A closer look at the data showed that a major problem in the implementation of the annotation is the number of sentences and words used within a comment. About 25% of the comments had plenty of long sentences. On the remaining 75% of the data a global agreement of $\alpha'_{global} \approx 0.6$ could be measured. For the two groups the agreement with $\alpha'_{control} \approx 0.315$ as well as $\alpha'_{discuss} \approx 0.524$ could thus be established. Despite a correction phase, small IAA values and high complexity, similar outcomes were determined as sufficiently good by [12] for a comparable annotation task.

Figure 4 shows the development of the IAA value on the non-adjusted data. All comments with a minimum number of words or sentences are examined. Both figures show that α falls with an increasing number of words or sentences per comment. It is clear that the IAA is high until more comments with plenty of long sentences are included. Exceeding the word and sentence boundaries leads to a divergence in the annotators views. While the agreement for the control-group oscillates around the expected value $\alpha_{expected} \approx 0.3$, it is clear that the data generated in discuss always produces a high degree of agreement regarding the recognition of the arguments it contains. Since discuss specifies a pattern for the input, arguments and their structure can be better recognized. In comparison with the control-group, which does not provide such a structure, the results obtained by discuss are better with regard to α. Therefore, the decrease of the α value with respect to discuss could be explained by the fact that by adding more long sentences it is no longer possible to distinguish between the different forms of arguments as described by [13]. Nevertheless, it turns out that the structure is essentially involved in the interpretation of arguments. Thus, a dialog-based system, like discuss, supports this understanding better than a simple system.

5. Related Work

Several experiments researching effects of online discussions have been conducted. Lampe et al. [14] researched how civility in online discussions is affected by choice of moderation system. They used the forums of *Slashdot* as a control for civil discussions and moderation. Another field-study by Rhee and Kim [15] tested whether online discussions could change the quality of a deliberative process. However, they conducted their experiments on the Internet and not in a lab setting. A study similar to this paper was conducted by Iandoli et al. [16]. They pit their collaborative online discussion tool against conventional forums, which are threaded and comment based. Here an online political process of an Italian party was used instead of a controlled lab setting.

Other studies utilized dialog-based discussions in their experiments as well. Krauthoff et al. [17] conducted a study where more than 300 students participated online in a discussion regarding the betterment of a study course. In contrast to this work no control-group was used. Another experiment utilizing dialog-based discussions was done by Ebbinghaus [18,19]. In this case the test was more geared toward whether decision-making processes based on dialog-based discussions are viable.

6. Conclusion

In this paper we presented a hypotheses-guided lab experiment to compare traditional comment-style online-argumentation and a dialog-based alternative. The main results show that participants prefer using the traditional comment-style methods and are overall more accustomed to them. We argued that improving the design of the user interface and improving the user experience should be a top priority in order to change this. As a second key result our data shows that the dialog-based approach leads the participants to produce both significantly more arguments and better structured arguments. It is therefore clearly worthwhile to put effort into improving the user interface and the user experience.

A secondary result is that experiments, as described in this paper, are a valuable tool to understand what the real-world problems and benefits of a proposed online argumentation approach are. In order to concentrate our efforts on the bottleneck issues and not on side issues it makes a lot of sense to use them more often. In the following we therefore outline how to improve the experiments themselves.

Since our participants were predominantly students, one could assume a certain bias towards discussion affinity. One way to improve the results would therefore be to conduct tests with subjects that conform to an intersection of the general populace. Furthermore, repeating the study with different articles and the same participants could give us a hint whether familiarization with the software might increase the usability ratings.

Another aspect that might be worthwhile to look into more detail is the annotation of arguments contained in the comments. We followed a very open definition since no participant was trained in formal argumentation. It would be interesting to see how the annotations change, when more restrictive definitions of argument are used.

It would also be very interesting to repeat the study with slight variations to determine the impact of those variations on the outcome. For example, using a dialog-based system with a different interface than discuss could show whether the lower usability ratings are inherent to dialog-based approaches or are caused by specific implementation details. A repeat-study with a less controversial topic could help to understand whether comment based approaches fare better, when there is less need for argumentation.

Acknowledgements Thanks to Dennis Frieß and his support to prepare the study. He also pointed us to some papers giving us a better understanding on conducting studies.

September 2020

References

[1] C. Meter, T. Krauthoff, and M. Mauve, “discuss: Embedding dialog-based Discussions into Websites,” in *Human Computer Interaction International: Proceedings of HCI*, 2017.

[2] T. Krauthoff, C. Meter, G. Betz, M. Baurmann, and M. Mauve, “D-BAS – A Dialog-Based Online Argumentation System,” in *Computational Models of Argument*, September 2018, pp. 325–336.

[3] A. Schneider and C. Meter, “Various efforts of enhancing real world online discussions,” in *ECA 2019: Proceedings of the 3rd European Conference on Argumentation*, June 2019.

[4] R. Likert, “A technique for the measurement of attitudes.” *Archives of psychology*, 1932.

[5] O. R. Holsti, “Content analysis for the social sciences and humanities,” *Reading. MA: Addison-Wesley (content analysis)*, 1969.

[6] A. Mullick, S. Ghosh, R. Dutt, A. Ghosh, and A. Chakraborty, “Public sphere 2.0: Targeted commenting in online news media,” in *European Conference on Information Retrieval*. Springer, 2019, pp. 180–187.

[7] R. Artstein, “Inter-annotator agreement,” in *Handbook of Linguistic Annotation*. Springer Netherlands, 06 2017, pp. 297–313.

[8] M. Ziegele, T. Breiner, and O. Quiring, “What creates interactivity in online news discussions? an exploratory analysis of discussion factors in user comments on news items,” *Journal of Communication*, vol. 64, no. 6, pp. 1111–1138, 2014.

[9] N. Springer, I. Engelmann, and C. Pfaffinger, “User comments: Motives and inhibitors to write and read,” *Information, Communication & Society*, vol. 18, no. 7, pp. 798–815, 2015.

[10] K. Krippendorff, “Content analysis: An introduction to its methodology.” Sage Publications, 2004, vol. 2, pp. 241–s243.

[11] ——, “Computing Krippendorff’s Alpha-Reliability,” https://repository.upenn.edu/asc_papers/43/, 2008, accessed: 2020-02-21.

[12] F. Boltuzic and J. Šnajder, “Back up your stance: Recognizing arguments in online discussions,” 01 2014, pp. 49–58.

[13] J. Lawrence and C. Reed, “Argument mining: A survey,” *Computational Linguistics*, pp. 1–55, 10 2019.

[14] C. Lampe, P. Zube, J. Lee, C. H. Park, and E. Johnston, “Crowdsourcing civility: A natural experiment examining the effects of distributed moderation in online forums,” *Government Information Quarterly*, vol. 31, no. 2, pp. 317–326, 2014.

[15] J. W. Rhee and E.-m. Kim, “Deliberation on the net: Lessons from a field experiment,” *Online deliberation: Design, research, and practice*, pp. 223–232, 2009.

[16] L. Iandoli, I. Quinto, P. Spada, M. Klein, and R. Calabretta, “Supporting argumentation in online political debate: Evidence from an experiment of collective deliberation,” *new media & society*, vol. 20, no. 4, pp. 1320–1341, 2018.

[17] T. Krauthoff, C. Meter, and M. Mauve, “Dialog-Based Online Argumentation: Findings from a Field Experiment,” in *Proceedings of the 1st Workshop on Advances in Argumentation in Artificial Intelligence*, vol. 2012. CEUR-WS, 2017, pp. 85–99.

[18] B. Ebbinghaus, “Decision Making with Argumentation Graphs,” Master’s thesis, Department of Computer Science, Heinrich-Heine-University Düsseldorf, May 2019.

[19] B. Ebbinghaus and M. Mauve, “decide: Supporting participatory budgeting with online argumentation,” in *Submitted to Computational Models of Argument*, 2020, under review.

Chapter 8

Conclusion and Future Work

In this Chapter, I want to summarize the contents of this thesis, give an outlook to future research and software development based on the work done here and end with a few personal closing words.

8.1 Conclusion

Discussions on the internet largely followed the same forms for a long time. Approaches to establish more formalized software to support and conduct discussions were typically aimed at enthusiast and expert users. Dialog-based discussions in turn aimed at laymen users as well. But like other forms, they had a range of open challenges. In this dissertation we proposed distributed arguments to tackle some of those challenges.

As a first step we could show that distributed networks, where the peers only accept data that fits their preferences, are functionally viable [35]. Although this may not seem to be trivially connected to the field of discussion software, it laid the groundwork for a distributed architecture which supports discussions and arguments that span over multiple entities. Any participating server can chose its own set of discussions and arguments to host, and the network will still be functional.

Building on that knowledge we developed and discussed a set of challenges which needed to be researched and solved to satisfactorily implement distributed argument networks [36]. Especially the context-dependence and the update-ability of arguments in a distributed environment were challenges, that were novel in this context. The result of solving those challenges was an architecture we proposed. We showed how it tackles the previously postulated challenges and provided a reference implementation – EDEN [26]. With EDEN, it is possible for any provider of content, e.g. forums, social media, news, etc. to become part of a decentralized argumentation network. The arguments are automatically synchronized between the providers, while a community-based mechanism filters statements that are against the rules. We solved the problem of decentralized argument updates by using a system similar to source code versioning. To sum up, we provided the first user-friendly suite to be a provider inside the argumentation network.

Building on that, we presented a suite of dialog-based tools, that can be used in conjunction to create a software pipeline for practical user-focused online argumentation. We introduced the tools and discussed the interplay between them to showcase an exemplary setup [37].

To evaluate the work we have done, we conducted a field-study, which was supposed to test how our solutions fare against established comment-style discussions [38]. We concluded that comment-style solutions are easier to understand and use, due to familiarity. On the other hand, we could show that our solutions entice users to produce more than double the amount of arguments, which clearly shows that our systems enable objective and fact-based discussions.

Over the span of four years that it took to finish this dissertation, we set out to create a practical solution for decentralized discussions, as to not rely on centralized services and destroy adverse effects like online echo-chambers. The end-result with all its many steps in between is EDEN and its accompanying ecosystem.

8.2 Future Work

Based on the knowledge and experiences gained from this dissertation, there are a few reasonable paths for future work.

The most obvious one would be to work on a more user-friendly and intuitive interface for distributed dialog-based argumentation. The experiment we conducted shows that the biggest detriment of our approach in comparison to conventional comments is the interface we used. An enhanced interface could lead to higher adoption rates, while retaining or even bettering the rate of arguments created per user.

Building more on EDEN, it would be of significance to incorporate automated argument and provider discovery. Currently, new providers are discovered by adding them manually to the whitelist and through the import of arguments of yet unknown providers. Further discovery mechanisms would immensely help to interconnect the global argument graph. Based on a tightly interconnected graph, there could be an automated scraper and "search engine" for arguments. It should be fairly doable to create a web-service that searches the global argument network for arguments and indexes them. Such a service would help to construct informed arguments, as well as to find discussions regarding topics of interests.

One point for a major improvement of EDEN in particular would be a completely automated import of whole sub-discussions without manual intervention. Currently, it is proposed that a review-system based on the wisdom of the crowd judges whether potentially fitting arguments should be imported or not. An AI, which judges whether arguments are context-free could fully automate this and thus enable the import of complete sub-trees without delay. This would help the user-experience and heighten the interconnections of the global argument graph.

Last but not least, we envisioned that user-profiles including their favorite discussions, sessions and saved arguments could be distributed along the arguments in the network. Making this possible in satisfactorily manner would probably take the effort of at least a dissertation in itself, but make systems in the argument network more desirable for end-users at once, since the hassle of keeping up with several providers would vanish completely.

8.3 Closing Words

During the last years, we explored the intersection between distributed computing and dialog-based argumentation – an intersection almost unexplored before. We learned that we can help guide users to pursue more factual and objective discussions. We saw that treating arguments and their interconnections as resources added value to discussions. And we saw that we are just at the beginning of the distributed argumentation journey.

When we look at the discussions on the Internet today, we see a landscape dominated by a cacophony of fake news, personal attacks, trolls and other disruptive factors. We do not claim that we can solve all of those problems easily, but we are optimistic that we as a community make steps in the right direction. We aim towards a future where a network of arguments is as prevalent as the Internet today. For a world, where people discuss topics to gain new perspectives and not to confirm their preconceived notions. For a world, where people seek further understanding of each other and not more reasons for division.
There is still much work to do.

Acronyms

P2P Peer-to-Peer

EDEN Extensible Discussion Entity Network

D-BAS Dialog-Based Argumentation System

Bibliography

[1] Zane L Berge. "Electronic discussion groups". In: *Communication Education* 43.2 (1994), pp. 102–111 (Page: 1).

[2] Floris Bex, John Lawrence, Mark Snaith, and Chris Reed. "Implementing the argument web". In: *Communications of the ACM* 56.10 (2013), pp. 66–73 (Page: 11).

[3] Floris Bex, Mark Snaith, John Lawrence, and Chris Reed. "Argublogging: An application for the argument web". In: *Journal of web semantics* 25 (2014), pp. 9–15 (Page: 11).

[4] Carlos Chesnevar, Sanjay Modgil, Iyad Rahwan, Chris Reed, Guillermo Simari, Matthew South, Gerard Vreeswijk, Steven Willmott, et al. "Towards an argument interchange format". In: *The knowledge engineering review* 21.4 (2006), pp. 293–316 (Page: 11).

[5] disqus.com. *Disqus - The #1 way to build an audience on your website*. 2020. URL: https://disqus.com/ (visited on 01/30/2020) (Page: 10).

[6] Maeve Duggan and Aaron Smith. "6% of online adults are reddit users". In: *Pew Internet & American Life Project* 3 (2013), pp. 1–10 (Page: 10).

[7] Seth Flaxman, Sharad Goel, and Justin M Rao. "Filter bubbles, echo chambers, and online news consumption". In: *Public opinion quarterly* 80.S1 (2016), pp. 298–320 (Page: 4).

[8] Kalman Graffi. "PeerfactSim. KOM: A P2P system simulator—Experiences and lessons learned". In: *2011 IEEE International Conference on Peer-to-Peer Computing*. IEEE. 2011, pp. 154–155 (Page: 14).

[9] Christian Grobe. "The power of words: Argumentative persuasion in international negotiations". In: *European Journal of International Relations* 16.1 (2010), pp. 5–29 (Page: 1).

[10] Anthony Hunter, Lisa Chalaguine, Tomasz Czernuszenko, Emmanuel Hadoux, and Sylwia Polberg. "Towards Computational Persuasion via Natural Language Argumentation Sialogues". In: *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*. Springer. 2019, pp. 18–33 (Page: 1).

[11] Ian Hutchby. "Power in discourse: The case of arguments on a British talk radio show". In: *Discourse & Society* 7.4 (1996), pp. 481–497 (Page: 1).

[12] Constanza Ihnen Jory et al. *Pragmatic argumentation in law-making debates: Instruments for the analysis and evaluation of pragmatic argumentation at the Second Reading of the British Parliament*. AmsterdamSic Sat, 2012 (Page: 1).

[13] Kialo Inc. *kialo - Explore Popular Debates, Discussions and Critical Thinking...* 2020. URL: https://www.kialo.com/ (visited on 02/07/2020) (Page: 10).

[14] Reddit Inc. *reddit: The front page of the Internet*. 2020. URL: https://www.reddit.com/ (visited on 01/30/2020) (Pages: 3, 10).

[15] Alan B Johnston and Daniel C Burnett. *WebRTC: APIs and RTCWEB protocols of the HTML5 real-time web*. Digital Codex LLC, 2012 (Page: 13).

[16] Tim Kelly and Rob Weaver. "The goal structuring notation–a safety argument notation". In: *Proceedings of the dependable systems and networks 2004 workshop on assurance cases*. Citeseer. 2004, p. 6 (Page: 1).

[17] Mark Klein. "How to harvest collective wisdom on complex problems: An introduction to the mit deliberatorium". In: *Center for Collective Intelligence working paper* (2011) (Page: 11).

[18] Tobias Krauthoff, Michael Baurmann, Gregor Betz, and Martin Mauve. "Dialog-Based Online Argumentation." In: *COMMA*. 2016, pp. 33–40 (Pages: 3, 10).

[19] Tobias Krauthoff, Christian Meter, and Martin Mauve. "Dialog-Based Online Argumentation: Findings from a Field Experiment." In: *AI^3 @ AI* IA*. 2017, pp. 85–99 (Page: 59).

[20] Travis Kriplean, Jonathan Morgan, Deen Freelon, Alan Borning, and Lance Bennett. "Supporting reflective public thought with considerit". In: *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*. 2012, pp. 265–274 (Page: 11).

[21] John Lawrence, Floris Bex, and Chris Reed. "Dialogues on the Argument Web: Mixed Initiative Argumentation with Arvina." In: *COMMA*. 2012, pp. 513–514 (Page: 11).

[22] John Lawrence, Floris Bex, Chris Reed, and Mark Snaith. "AIFdb: Infrastructure for the Argument Web." In: *COMMA*. 2012, pp. 515–516 (Page: 11).

[23] Neil MacCormick. "Argumentation and interpretation in law". In: *Ratio Juris* 6.1 (1993), pp. 16–29 (Page: 1).

[24] Petar Maymounkov and David Mazieres. "Kademlia: A peer-to-peer information system based on the xor metric". In: *International Workshop on Peer-to-Peer Systems*. Springer. 2002, pp. 53–65 (Page: 14).

[25] Christian Meter, Tobias Krauthoff, and Martin Mauve. "discuss: Embedding dialog-based Discussions into Websites". In: *International Conference on Learning and Collaboration Technologies*. Springer. 2017, pp. 449–460 (Pages: 5, 10).

[26] Christian Meter, Alexander Schneider, and Martin Mauve. "EDEN: Extensible Discussion Entity Network." In: *COMMA*. 2018, pp. 257–268 (Pages: 31, 75).

[27] R L Morgan, Scott Cantor, Steven Carmody, Walter Hoehn, and Ken Klingenstein. "Federated security: The shibboleth approach". In: *Educause Quarterly* 27.4 (2004), pp. 12–17 (Page: 9).

[28] Tiago Oliveira, Jérémie Dauphin, Ken Satoh, Shusaku Tsumoto, and Paulo Novais. "Argumentation with Goals for Clinical Decision Support in Multimorbidity". In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. 2018 (Page: 1).

[29] Sheizaf Rafaeli. "The electronic bulletin board: A computer-driven mass medium". In: *Social Science Micro Review* 2.3 (1984), pp. 123–136 (Page: 1).

[30] Iyad Rahwan and Chris Reed. "The argument interchange format". In: *Argumentation in artificial intelligence*. Springer, 2009, pp. 383–402 (Page: 11).

[31] Iyad Rahwan, Fouad Zablith, and Chris Reed. "Laying the foundations for a world wide argument web". In: *Artificial intelligence* 171.10-15 (2007), pp. 897–921 (Page: 11).

[32] Chris Reed and Glenn Rowe. "Araucaria: Software for argument analysis, diagramming and representation". In: *International Journal on Artificial Intelligence Tools* 13.04 (2004), pp. 961–979 (Page: 11).

[33] Eugen Rochko. *Giving Social Networks back to you - Mastodon*. 2020. URL: https://joinmastodon.org/ (visited on 01/15/2020) (Page: 9).

[34] Peter Saint-Andre. “Streaming xml with jabber/xmpp”. In: *IEEE internet computing* 9.5 (2005), pp. 82–89 (Page: 9).

[35] Alexander Schneider and Martin Mauve. “Compliance management for P2P systems”. In: *Communications (APCC), 2017 23rd Asia-Pacific Conference on*. ©2017 IEEE. 2017, pp. 1–6 (Pages: 13, 75).

[36] Alexander Schneider and Christian Meter. “Reusable Statements in Dialog-Based Argumentation Systems.” In: *AI^ 3@ AI*IA*. 2017, pp. 100–104 (Pages: 23, 75).

[37] Alexander Schneider and Christian Meter. “Various Efforts of Enhancing Real World Online Discussions”. In: *Proceedings of the 3rd European Conference on Argumentation, Volume III*. College Publications. 2020, pp. 311–323 (Pages: 47, 76).

[38] Alexander Schneider, Christian Meter, Marc Feger, Jan Steimann, and Martin Mauve. “discuss vs. Disqus: Evaluating Dialog-Based Discussions Against a Comment-Based System”. In: *Submitted to Computational Models of Argument: Proceedings of COMMA 2020 - in review during submission of this dissertation* (2020) (Pages: 59, 76).

[39] Christina Slade. “Reasons to buy: The logic of advertisements”. In: *Argumentation* 16.2 (2002), pp. 157–178 (Page: 1).

[40] Christine Tippett. “Argumentation: The language of science”. In: *Journal of Elementary Science Education* 21.1 (2009), pp. 17–25 (Page: 1).

Curriculum Vitae

I. Personal Details

Name	Alexander Schneider
Birth date	20.09.1990
Birth place	Krasnij Jar, Kasachstan
Academic degree	Master of Science
Nationality	German
Family Status	Married to Maria Schneider (née Kersten)
Personal Interests	Programming, Esports, Music
Languages	German, English, Russian, Spanish

Contact Data

Office Address	Room 25.12.02.41 Institute of Computer Science Heinrich Heine University Düsseldorf Universitätsstrasse 1 40225 Düsseldorf
Telephone	+49 211 81-11637
E-Mail	alexander.schneider@hhu.de
Homepage	schneider.gg

Academic and Professional Experience

Since 09/2017	Founder & CTO of artSafe GmbH
Since 07/2015	Researcher in the field of decentralized online argumentation at the Computer Networks chair at the Heinrich Heine University in Düsseldorf. The chair lead is Prof. Dr. Martin Mauve.
07/2013 - 07/2015	Master's Degree in Computer Science at Heinrich Heine University Düsseldorf. Thesis: "Attacks against Network Voting Systems" (Grade: 1.0) Supervisors: Prof. Dr. Martin Mauve, Jun.-Prof. Dr.-Ing. Kalman Graffi Advisor: Philipp Hagemeister, M.Sc. Degree Grade: 1.0
10/2010 - 07/2013	Bachelor's Degree in Computer Science at Heinrich Heine University Düsseldorf. Thesis: "Improving Dependency Resolution of Python Packages" (Grade: 1.0) Supervisors: Prof. Dr. Martin Mauve, Prof. Dr. Michael Leuschel Advisor: Philipp Hagemeister, M.Sc. Degree Grade: 1.5
07/2009 - 08/2010	Civil service at the protestant church in Wegberg
1996 - 2009	Elementary, secondary and high school Abitur Degree "Allgemeine Hochschulreife"

II. Teaching

Lectures

SS 20

Lecture: "Introduction to Computer Networks, Databases and Operating Systems" at Heinrich Heine University Düsseldorf.
Topic: Introduction to need-to-know concepts in computer science.
Responsibility: Co-Coordination of exercises and intermediate tests.

WS 19/20

Lecture: "Advanced Programming and Algorithms" at Heinrich Heine University Düsseldorf.
Topic: Introducing data scientists to algorithmic and programming concepts in Python.
Responsibility: Coordination and Execution of programming exercises.

WS 18/19

Lecture: "Computer Networks" at Heinrich Heine University Düsseldorf.
Topic: Layers of typical networks, communication between computing devices.
Responsibility: Coordination and Execution of theoretical and programming exercises. Giving some lectures.

SS 18

Lecture: "Network-Security" at Heinrich Heine University Düsseldorf.
Topic: Overview of common attack and defense patterns in networking and web security.
Responsibility: Giving the lecture.

WS 17/18

Lecture: "Online Participation for Argumentation Systems" at Heinrich Heine University Düsseldorf.
Topic: Online participation technology, online discussions, argumentation theory.
Responsibility: Co-Coordination and giving lectures.

SS 17

Lecture: "Network-Security" at Heinrich Heine University Düsseldorf.
Topic: Overview of common attack and defense patterns in networking and web security.
Responsibility: Organizing and giving the practical exercises.

WS 16/17 Lecture: “Computer Networks” at Heinrich Heine University Düsseldorf.
Topic: Layers of typical networks, communication between computing devices.
Responsibility: Coordination and Execution of theoretical and programming exercises. Giving some lectures.

WS 16/17 Lecture: “Introductory Pre-Course for Computer Science” at Heinrich Heine University Düsseldorf.
Topic: Everything needed for a computer scientist to start the first semester.
Responsibility: Course-Coordination.

SS 16 Lecture: “Network-Security” at Heinrich Heine University Düsseldorf.
Topic: Overview of common attack and defense patterns in networking and web security.
Responsibility: Organizing and giving the practical exercises.

WS 15/16 Lecture: “Computer Networks” at Heinrich Heine University Düsseldorf.
Topic: Layers of typical networks, communication between computing devices.
Responsibility: Co-Coordination and Execution of theoretical and programming exercises.

WS 11/12 - SS 15 Several Lectures in the Computer Science Course at Heinrich Heine University Düsseldorf.
Responsibility: Tutoring other students.

Theses supervised

WS 19/20 Bachelor’s Thesis from Elmedin Turic:
“Development of an Administration-Tool for VoIP Configurations.”

SS 19 Bachelor’s Thesis from Stephan Linzbach:
“Development of an Automated Wiki-Software for Argumentation-Networks.”

SS 19 Bachelor’s Thesis from Jan Schnorrenberg:
“Design and Implementation of a Relative Majority-Based Voting System Based on the Stellar Blockchain.”

WS 18/19 Bachelor's Thesis from Simon Weber:
"Security Analysis of Bluetooth Low-Energy Applications and Hardware."

WS 18/19 Bachelor's Thesis from Benedikt Peter Schmeitz:
"Development of a Trust-Based Product Recommendation-Network."

WS 18/19 Bachelor's Thesis from Frederik Maximilian Aulich:
"Automated Topic-Mining of German-Language Social Media Content."

SS 18 Bachelor's Thesis from Danny Nguyen:
"Analysis of Steering-Effects in Gamification of Online Discussion Software."

SS 18 Bachelor's Thesis from Alexander Disterhöft:
"Development of an Article Scraper for German Newspapers including Topic Analysis."

SS 18 Bachelor's Thesis from Steven Michel:
"Modernization of a Company-Network According to Modern Procedures."

SS 18 Bachelor's Thesis from Tolga Mizrak:
"Gamification of Online Discussion Software."

SS 18 Bachelor's Thesis from Patrick Koch:
"Design and Implementation of a Review-system for Dialog-based Discussions."

WS 17/18 Bachelor's Thesis from Jens Niklas Albert:
"Programatic Creation of a Realistic Discussion-Graph."

WS 17/18 Bachelor's Thesis from David Frederic Dankelmann:
"Development of an Android Application for a Keystroke-Injection via USB."

WS 17/18 Bachelor's Thesis from Julian Zenz:
"An Analysis of Modern Ransomware Based on the Vault 7 Leaks."

SS 17 Bachelor's Thesis from Nikita Hasert:
"Ad-Hoc Signaling in Web RTC."

III. Services for the Community

2020	Co-Creator of emergency backup SARS-CoV-2 / COVID-19 information website for Heinrich Heine University Düsseldorf.
Since 2018	Co-Organizer of rheinJUG, the Java User Group in Düsseldorf
2018	Volunteer at PhD ceremony of the natural sciences faculty at Heinrich Heine University Düsseldorf.
Since 2017	Co-Organizer of the Clojure Düsseldorf Meetup
2011-2016	Elected student body representative computer science at Heinrich Heine University Düsseldorf.
2012-2015	Treasurer at the computer science student body at Heinrich Heine University Düsseldorf.

IV. Work-Related Engagements

Since 2016	Founding Member of the Düsseldorf Institute for Internet and Democracy.
2016-2020	Associated Member of the Fortschrittskolleg Online-Participation NRW.
2018 - 2019	Elected to the scientific body of computer science faculty representing the scientific staff at Heinrich Heine University Düsseldorf.

Personal Publications

Reviewed Conference Papers

[P1] Alexander Schneider, Martin Mauve. "Compliance Management for P2P Systems". In: *Proceedings of the 23rd Asia-Pacific Conference on Communications (APCC)*. 2017.

[P2] Alexander Schneider, Christian Meter. "Reusable Statements in Dialog-Based Argumentation Systems". In: *CEUR Workshop Proceedings AI*IA Series (AI3)*. 2017.

[P3] Christian Meter, Alexander Schneider, Martin Mauve. "EDEN: Extensible Discussion Entity Network". In: *Computational Models of Argument (COMMA)*. 2018.

[P4] Alexander Schneider, Christian Meter. "Various Efforts of Enhancing Real World Online Discussions". In: *Proceedings of the 3rd European Conference on Argumentation (ECA)*. 2019.

[P5] Christian Meter, Alexander Schneider, Marc Feger, Jan Steimann, Martin Mauve "discuss vs. Disqus: Evaluating Dialog-Based Discussions Against a Comment-Based System" Under review in: *Computational Models of Argument (COMMA)*. 2020.

Articles

[A1] Christian Meter, Tobias Krauthoff, Alexander Schneider. "Dialogbasierte Online-Diskussionen". In: *Deutsche Verwaltungspraxis*. 2018.

Technical reports

[R1] Alexander Schneider, Christian Meter, Philipp Hagemeister. "Survey on Remote Electronic Voting". In: *arXiv.org, Cornell University Library*. 2017.

[R2] Christian Meter, Alexander Schneider, Philipp Hagemeister, Martin Mauve. "Tor is not enough: Coercion in Remote Electronic Voting Systems". In: *arXiv.org, Cornell University Library*. 2017.

Theses

[T1] Alexander Schneider. “Improving Dependency Resolution of Python Packages”. Bachelor’s thesis. *Heinrich Heine University, Düsseldorf, Germany.* August 2013.

[T2] Alexander Schneider. “Attacks against Network Voting Systems”. Masters’s thesis. *Heinrich Heine University, Düsseldorf, Germany.* August 2015.

Eidesstattliche Erklärung
laut §5 der Promotionsordnung vom 06.12.2013

Ich versichere an Eides Statt, dass die Dissertation von mir selbständig und ohne unzulässige fremde Hilfe unter Beachtung der „Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf“ erstellt worden ist.

Ort, Datum

Alexander Schneider

Please add here

the DVD holding sheet

This DVD contains:

- A *PDF* version of this thesis
- All LaTeXand grafic files that have been used, as well as the corresponding scripts
- The referenced websites and papers