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.