Various Efforts of Enhancing Real World Online Discussions

ALEXANDER SCHNEIDER
Heinrich-Heine-University Düsseldorf, Germany
alexander.schneider@hhu.de

CHRISTIAN METER
Heinrich-Heine-University Düsseldorf, Germany
christian.meter@hhu.de

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.

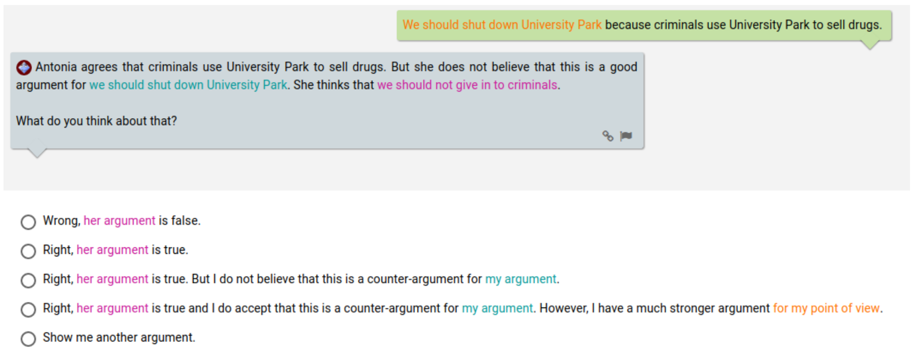
KEYWORDS: argumentation, online discussions, dialog-based discussions, web applications

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.


We should shut down University Park because criminals use University Park to sell drugs.

Antonia agrees that criminals use University Park to sell drugs. But she does not believe that this is a good argument for we should shut down University Park. She thinks that we should not give in to criminals.

What do you think about that?

- Wrong, her argument is false.
- Right, her argument is true.
- Right, her argument is true. But I do not believe that this is a counter-argument for my argument.
- Right, her argument is true and I do accept that this is a counter-argument for my argument. However, I have a much stronger argument for my point of view.
- Show me another argument.

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

A typical user is presented with an argument and the request to react to that argument (see Figure 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 deduct from the selected choices of the user.

In this paper we describe three such interfaces, namely our *Dialog-Based Argumentation System* (D-BAS) (Krauthoff et al., 2018), *discuss* (Meter et al., 2017), and *Jebediah* (Meter et al., 2018). The interfaces differ in their approaches and use-cases. 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 their backend, which is accessible via D-BAS' *Application Programming Interface* (API) in the reference implementations.

The structured data created by the interfaces lends itself to reuse, and as a consequence we also present our *Extensible Discussion Entity Network* (EDEN) (Meter et al., 2018). 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* (Neugebauer, 2018), which enables the transformation of the gathered data into instances of *Argumentation Frameworks* (AF) (Dung, 1995), *Abstract Dialectical Framework* (ADF) (Brewka & Woltran, 2010) and ASPIC+ (Caminada & Amgoud, 2007).

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 ONLINE 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 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 Online Argumentation

Dialog-based argumentation was introduced in detail by Krauthoff et al. (2016, 2018) 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.

Let us 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 4. 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.

Let us 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 mini- mum. 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 is 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.

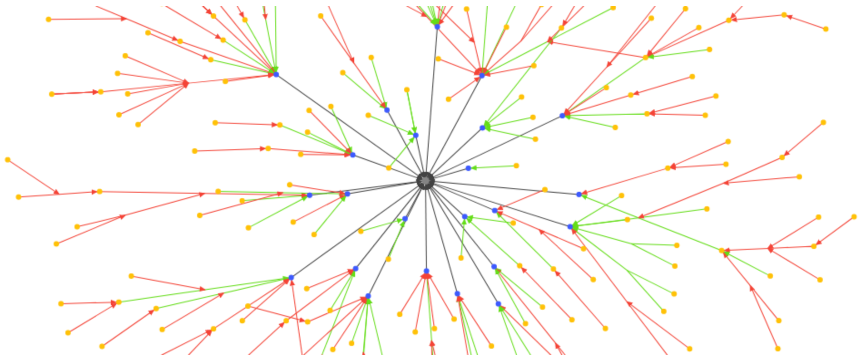


Figure 2: The graph resulting from discussions through D-BAS, discuss and Jebediah. Depicted is an instance from a real-world discussion. Colors: grey: issue, blue: positions, yellow: statements, green: supports, red: attacks.

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* (Krauthoff et al., 2018). 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[1]. Besides experiences gathered from running the service, there also have been lessons learned from a formal evaluation through a field-study (Krauthoff et al., 2017). 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

[1] https://dbas.cs.hhu.de

dialog-based argumentation. Parts of the resulting graph can be seen in Figure 2 and the associated data can be obtained online.

2.4 Application Programming Interfaces

D-BAS has two fully documented[2] 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) (Bex et al., 2014) 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.

Data retrieval from our databases can be achieved using the second endpoint, which provides a GraphQL (The GraphQL Foundation, 2019) 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 providing content and the space

[2] https://dbas.cs.hhu.de/docs

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 (2018) 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

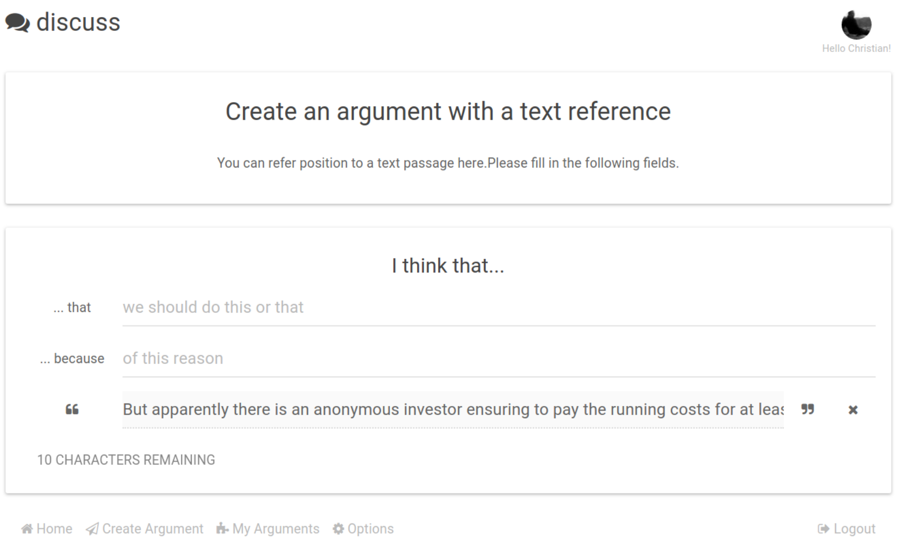
An exemplary implementation of a distributed argumentation network powered by aggregators is EDEN which was presented in detail by Meter et al. (2018). EDEN was developed in Clojure, a functional language on the JVM. Furthermore, we pursued a modular approach with EDENs architecture, which splits it 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 providing 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* (Meter et al., 2017). 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.



Create an argument with a text reference

You can refer position to a text passage here.Please fill in the following fields.

I think that...

... that we should do this or that

... because of this reason

But apparently there is an anonymous investor ensuring to pay the running costs for at leas

10 CHARACTERS REMAINING

Figure 3: discuss: Create a new argument with a reference to a passage in the author's article.

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 interacted with the article, (3) enabling discussions in our proposed dialog-based flow (see Subsection 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 Figure 3). 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 Figure 4). 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.

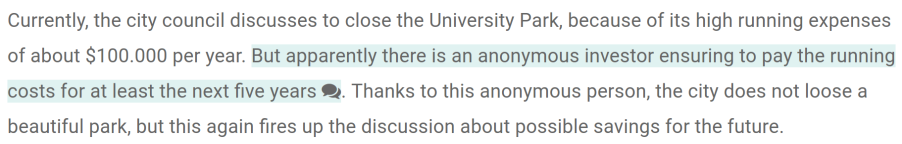

Currently, the city council discusses to close the University Park, because of its high running expenses of about $100.000 per year. But apparently there is an anonymous investor ensuring to pay the running costs for at least the next five years. Thanks to this anonymous person, the city does not loose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

Figure 4: Text passage from an article, which has been used in an argument. A click on it opens the interface to jump into the discussion

4.3 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 Subsection 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.

4.4 EDEN Integration

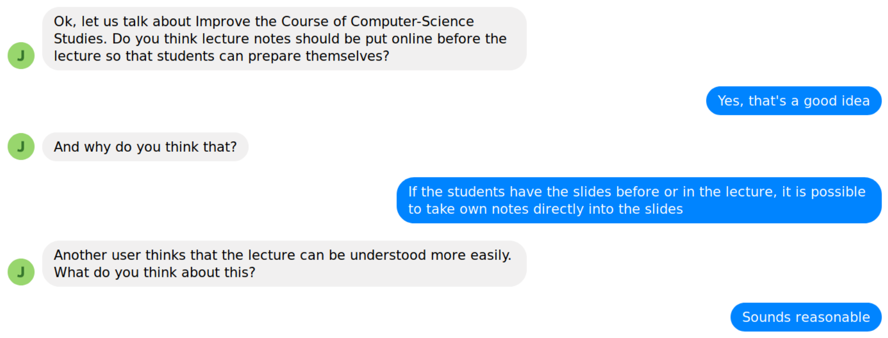
Besides the described functions, discuss can be used to connect to the EDEN network (see Subsection 3.2). D-BAS is then solely used as an DGEP for the steps in the discussions, whereas the arguments are being fetched from EDEN. This mechanism allows to retrieve and collect arguments from different locations 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 through 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 (Meter et al., 2018) 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 (Google Ireland Limited, 2019), 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 Figure 5). 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.



Ok, let us talk about Improve the Course of Computer-Science Studies. Do you think lecture notes should be put online before the lecture so that students can prepare themselves?

Yes, that's a good idea

And why do you think that?

If the students have the slides before or in the lecture, it is possible to take own notes directly into the slides

Another user thinks that the lecture can be understood more easily. What do you think about this?

Sounds reasonable

Figure 5: Left side: Dynamically produced text messages from Jebediah, right side the user's answers in the Facebook Messenger.

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* (Neugebauer, 2018). 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.

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 (Rahwan et al., 2007). 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.

(2012), 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 (Gordon & Walton, 2006), Deliberatorium (Klein & Iandoli, 2008) or OVA as introduced by Snaith et al. (2010). 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

Bex, F., Lawrence, J., & Reed, C. (2014, January). Generalising argument dialogue with the Dialogue Game Execution Platform. Frontiers in Artificial Intelligence and Applications 266, pp. 141-152.

Brewka, G., & Woltran, S. (2010). Abstract Dialectical Frameworks. Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning.

Caminada, M., & Amgoud, L. (2007). On the evaluation of argumentation formalisms. Artificial Intelligence, vol. 171, pp. 286-310.

Dung, P. (1995). 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.

Google Ireland Limited. (2019, June). Dialogflow. Retrieved from https://dialogflow.com/

Gordon, T., & Walton, D. (2006). The Carneades Argumentation Framework - Using Presumptions and Exceptions to Model Critical Questions. Frontiers in Artificial Intelligence and Applications, vol. 144: Computational Models of Argument, pp. 195-207.

Klein, M., & Iandoli, L. (2008, February 20). Supporting Collaborative Deliberation Using a Large-Scale Argumentation System: The MIT Collaboratorium. MIT Sloan Research Paper No. 4691-08.

Krauthoff, T., Baurmann, M., Betz, G., & Mauve, M. (2016). Dialog-Based Online Argumentation. Computational Models of Argument, pp. 33-40.

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, vol. 2012, CEUR-WS, pp. 85-99.

Krauthoff, T., Meter, C., Betz, G., Baurmann, M., & Mauve, M. (2018, September). D-BAS – A Dialog-Based Online Argumentation System. Computational Models of Argument, pp. 325-336.

Lawrence, J., Bex, F., Reed, C., & Snaith, M. (2012). AIFdb: Infrastructure for the Argument Web. Frontiers in Artificial Intelligence and Applications, vol. 144: Computational Models of Argument, pp. 515-516.

Meter, C., Ebbinghaus, B., & Mauve, M. (2018, September). Jebediah – Arguing With a Social Bot. Computational Models of Argument, pp. 467-468.

Meter, C., Krauthoff, T., & Mauve, M. (2017, July). discuss: Embedding dialog-based Discussions into Websites. Human Computer Interaction International: Proceedings of HCI, pp. 449-460.

Meter, C., Schneider, A., & Mauve, M. (2018, September). EDEN: Extensible Discussion Entity Network. Computational Models of Argument, pp. 257-268.

Neugebauer, D. (2018, September). DABASCO: Generating AF, ADF, and ASPIC instances from Real-World Discussions. Proceedings of the 2018 Conference on Computational Models of Argument, pp. 469-470.

Rahwan, I., Zablith, F., & Reed, C. (2007). Laying the foundations for a World Wide Argument Web. Artificial Intelligence, vol. 171, no. 10-15, pp. 897-921.

Snaith, M., Devereux, J., Lawrence, J., & Reed, C. (2010, August). Pipelining Argumentation Technologies. Proceedings of the 2010 conference on Computational Models of Argument, pp. 447-453.

The GraphQL Foundation. (2019, June). GraphQL - A query language for your API. Retrieved from https://graphql.org/