

---

# SEARCHING, NAVIGATING, AND QUERYING ARGUMENTS AND DEBATES: TOOLS, LANGUAGES AND METHODOLOGIES

GIORGOS FLOURIS

*Institute of Computer Science, FORTH, Heraklion, Greece.*  
fgeo@ics.forth.gr

JOHN LAWRENCE

*Centre for Argument Technology, University of Dundee, UK.*  
j.lawrence@dundee.ac.uk

DIMITRA ZOGRAFISTOU

*Centre for Argument Technology, University of Dundee, UK.*  
dzografistou@gmail.com

---

## Abstract

Recent developments in Web technologies have transformed Web users from passive consumers to active creators of digital content. As users see the Web as a means to enable dialogical exchange, debating, and commenting on products, services or events, a significant portion of web content is of argumentative form. This content can be unstructured, e.g., free-text, or (semi-)structured, both at the debate level, e.g., through a reply structure, and/or at the argument level, e.g., by requiring a specific argument format. The plethora of arguments online are useful only with the support of appropriate tools for identifying relevant arguments for any given information need. Depending on the form of the debate, the context, and the application at hand, this identification could require capabilities for simple keyword-based searching, navigational or explorational capabilities, the ability to perform analytical queries, as well as the ability to perform more complex searches and queries that involve the arguments' (or debates') structure and interrelationships. In this chapter, we provide a short survey of various tools and languages for searching, exploring and querying arguments, aiming to highlight the main advances in this area.

## 1 Introduction

The Web has long now ceased to be a purely one-directional means of communication: Web users are no longer just passive consumers of information, but can also contribute actively to the content of the Web. As a matter of fact, modern Web allows users to create and post different forms of digital information, such as the uploading of images and videos in social media channels; the posting of reviews for all types of products or services; the commentary of articles, political ideas or other viewpoints; the expression of ideas in chat rooms; the debating for hot controversial issues in specialised websites; and others.

One of the important consequences of this transition is that users often see the Web as a means to enable dialogical exchange, debating, and commenting, as it allows their ideas to reach people in all corners of the world. As a result, a significant portion of Web content is of argumentative form, containing users' opinions on any conceivable topic, often with well-articulated arguments. Depending on the platform in which these arguments are expressed, they can be unstructured (e.g., free-text), or have some kind of structure, which is imposed by the tool. This structure could be imposed at the debate level, e.g., through replies that end up generating debates that have a tree-like or forest-like structure, or at the argument level, by requiring the users to formulate their arguments in a specific way, e.g., explicitly specifying (or annotating) the premises and conclusions of the arguments.

The plethora of arguments online are useful only with the support of appropriate tools for identifying relevant arguments for any given information need. Note that keyword-based searching is useful but not sufficient for supporting the information needs of users when it comes to argumentative content. Arguments have an inherent structure (whether explicitly specified or not) and they are related to each other (although such relationships may not be always obvious and/or explicitly recorded); these properties are often critical and should be queryable, so whatever method we use for searching arguments should include the ability to perform more complex searches and queries that involve the arguments' (or debates') structure and interrelationships. Moreover, depending on the form of the debate, the context, and the application at hand, argument searching should also allow navigational

or explorational capabilities as well as the ability to perform analytical queries.

In this chapter, we provide a short survey of various approaches for searching, exploring and querying arguments, aiming to highlight the main advances in this area. We classify the relevant approaches into three main categories, namely, *query languages*, *argument retrieval systems*, and *closed debating systems*, which we explain below and in their respective sections.

Section 2 deals with the first category, and discusses structured query languages that one could use to query arguments, in a manner similar to other structured languages such as SQL or SPARQL. This approach is mainly applicable to arguments whose structure and/or interrelationships are explicitly recorded using an appropriate knowledge representation formalism, such as an ontology. Towards this direction, we present the only (to the best of our knowledge) structured query language that was designed explicitly for argumentative content and addresses these issues, namely *Argumentation Query Language (ArgQL)* [Zografistou *et al.*, 2017], [Zografistou *et al.*, 2018], [Roussakis *et al.*, 2022]. ArgQL abstracts argument/debate retrieval operations from the implementation details of the underlying repository, allowing the user to express queries in a more “natural” way (using argumentative terminology), leading to queries that are short, easy to formulate, read, understand and maintain, as well as efficient.

The second approach (argument retrieval systems) is discussed in Section 3. Argument retrieval systems are used to search for arguments in a manner analogous to how popular search engines crawl and search the Web for textual content. Argument retrieval systems can retrieve arguments that answer questions like “is coffee good for you?”, by looking at the argumentative content of various websites that have been previously analysed. To do so, such systems often use argument mining techniques for identifying the argumentative content of websites, and the structure/interrelationships of the contained arguments. In other cases, the arguments’ structure and interrelationships may have become known to the argument retrieval system because this structure is somehow exposed by the respective website, or because it has been identified through manual annotation.

The third approach (closed debating systems, analysed in Section 4) includes tools whose main functionality is the management and/or analysis of debates that have been created *within* the tool. In this respect the system is “closed”, because it provides a complete set of functionalities for both entering and retrieving arguments. Note how this contrasts with the argument retrieval systems, which provide searching, analytics and exploration capabilities for arguments expressed elsewhere. The fact that closed debating systems fully control the way in which the arguments are entered and stored in the underlying repository allows them to provide sophisticated searching and visualisation functionalities not easily attainable with other methods. On the downside, the pool of arguments over which searching is allowed is limited to the ones that have been ingested through the tool’s interface, which often limits their applicability to well-focused, small-scale debates within well-defined groups (e.g., among people involved in the management of a company pondering on a specific business decision).

The rest of this chapter presents further details on various representatives of these three approaches to argument searching (in their respective sections). We conclude in Section 5 with a general discussion and outlook for future research directions on the topic.

## 2 Languages for querying arguments

As mentioned above, searching and navigating within dialogues introduce specific challenges and requirements. This is mainly due to the fact that, at the core of debating and argumentation, particular structures are created, which are common in any kind of discussion regardless of its topic, the structure of which is queryable. This sets new ground in the area of information retrieval, in which new theories and models can be developed.

One such direction is the identification of the different kinds of queries related to the problem of dialogue searching and the exploration of the different ways in which they can be expressed, evaluated or executed more efficiently. The most obvious and straightforward approach is to adopt some of the standard storage schemes to represent dialogical data, and then use the associated query language to express requests for

particular fragments of the data [Cabrio *et al.*, 2013], [Cimiano *et al.*, 2013]. Such storage schemes include relational databases (MySQL), semantic databases (RDF/SPARQL), graph databases (GraphQL, Neo4j) etc. For all those frameworks, dialogical searching constitutes an application domain, using an appropriate knowledge representation formalism, such as an ontology. The most popular approach in this respect is the use of the *Argument Interchange Format (AIF)* [Chesñevar *et al.*, 2006] a data model that was designed to bridge the various models of argumentation into a common ontological pattern of representation. Although the existence of this de-facto representation standard allows the use of a general-purpose query language (such as SPARQL) for querying the argumentative data, this approach has several disadvantages. First, it requires the user to deeply understand AIF in order to properly formulate the correct query. Second, the resulting queries are often long and complex, thus error-prone and hard to maintain. Third, this approach creates an undesirable bonding between the query formulation process and the implementation details of the argumentation repository, reducing robustness and interoperability, as the application logic is bound to the specific implementation and thus cannot be easily migrated to alternative implementations.

The existence of a high-level language that uses terminology and semantics related to the argumentation domain, would offer to the community a more familiar way to express queries, saving them from dealing with all the technical details of using standard technologies. Furthermore, there is a large volume of dialogical and argumentative data distributed across various platforms and represented in different formats, and the development of centralised mechanisms to search through all this data, facilitating their integration, would enable the development of techniques for the automatic analysis of dialogues and debates. The only known language that carries the momentum of becoming such a language is *ArgQL (Argumentation Query Language)* [Zografistou *et al.*, 2017], [Zografistou *et al.*, 2018], [Roussakis *et al.*, 2022].

## 2.1 Argumentation Query Language (ArgQL)

ArgQL is a high-level, declarative query language, designed to express queries particularly related to the problem of searching within dialogues.

Its specification is designed on top of a data model that captures the fundamental principles of structured argumentation, and thus, data comprise graphs of interlinked, structured arguments. ArgQL enables querying both the internal structure of arguments and their relations, but also the structure of the debate that the arguments are part of. The debate structure can be seen in terms of Dung's argument frameworks [Dung, 1995], where arguments and relations are abstracted away from their structure, and define graphs where nodes represent arguments and edges the relations between them. By allowing querying such graphs, ArgQL makes itself compatible with those frameworks, and in future extensions, it could incorporate semantics of abstract frameworks in the search mechanisms.

ArgQL offers a simple and clear way to express queries that fall into four main categories (or their combinations): a) Locating individual arguments, e.g. *"Find arguments with conclusion p"* b) Identifying commonalities between arguments' structure, e.g. *"Find arguments with common premises"* c) Extracting argument relations, e.g. *"Find arguments that attack or support each other"* and d) Navigating within the argument graph e.g. *"Find arguments that attack the attackers of argument a"*. In all of these categories, special emphasis is given in the factor of rephrasing, or otherwise propositional equivalence, namely the fact that two propositions might be saying the same thing (be equivalent), but expressed in a different way. In particular, assuming that arguments consist of propositions, a query searches for propositions, that either themselves or any of their equivalent ones are satisfying a particular expression.

The current implementation of ArgQL is built on top of the RDF storage scheme. In particular, in order for a query to be executed, it is first translated into SPARQL, which is then executed against an RDF database, to which the data model of ArgQL has been mapped. The RDF schema that is used is specified by the AIF ontology [Chesñevar *et al.*, 2006]. In a similar way, more implementations of ArgQL can be developed, that will translate into other query languages, allowing that way for ArgQL to become a centralised language that integrates the results of querying different databases and heterogeneous argumentative data.

### 2.1.1 Data Model

The main concept in the data model of ArgQL is the *argument*, which consists of *propositions*. There are three kinds of relations between propositions, *equivalence*, *conflict* and *inference*. Equivalence and conflict relations define equivalence and contrariness in the content among propositions and are defined in a symmetric way. An argument is defined as a tuple  $\langle pr, c \rangle$ , where *pr* is a set of propositions, called *premise*, *c* is a single proposition, called *conclusion* and it holds that the premise logically infers the conclusion. A conflict relation between propositions defines two types of *attack* between arguments: *rebut* (conflict between two conclusions) and *undercut* (conflict between a conclusion and a premise). An equivalence relation between propositions defines two types of *support* between arguments: *endorse* (equivalent conclusions) and *backing* (equivalence between a conclusion and a premise). Overall, an argument base forms a graph, in which nodes are structured arguments, connected via four types of relations.

Figure 1 gives an example of the data model. Subfigures (a) and (b) show arguments' and relations' internal structure, while (c) depicts the structure of the debate. The nodes in this last graph view (c) represent an abstract version of arguments, the internal structure of which has the general form  $\langle pr, c \rangle$ , where *pr* infers *c*.

### 2.1.2 ArgQL specification

The general form of an ArgQL query is:

$$q \leftarrow \mathbf{match} \text{ dialogue\_pattern } (', ' \text{ dialogue\_pattern})^* \\ \mathbf{return} \text{ varlist } \mid \mathbf{path}(v_1, v_2)$$

where  $\text{varlist} = (v_1, v_2, \dots, v_n)$  is a list of variables. A *dialogue pattern* may have one of the following two forms:

$$\text{dialogue\_pattern} ::= \text{argpattern} \mid \\ \text{argpattern pathpattern dialogue\_pattern}$$

*Argument patterns* are the primary units in the language and are used to match arguments' internal structure. Syntactically, an argument pattern can either be a single variable  $v_a$ , or have the form  $v_a:\langle \text{premisePattern}, \text{conclusionPattern} \rangle$ . *PremisePattern* and *conclusionPattern* specify the

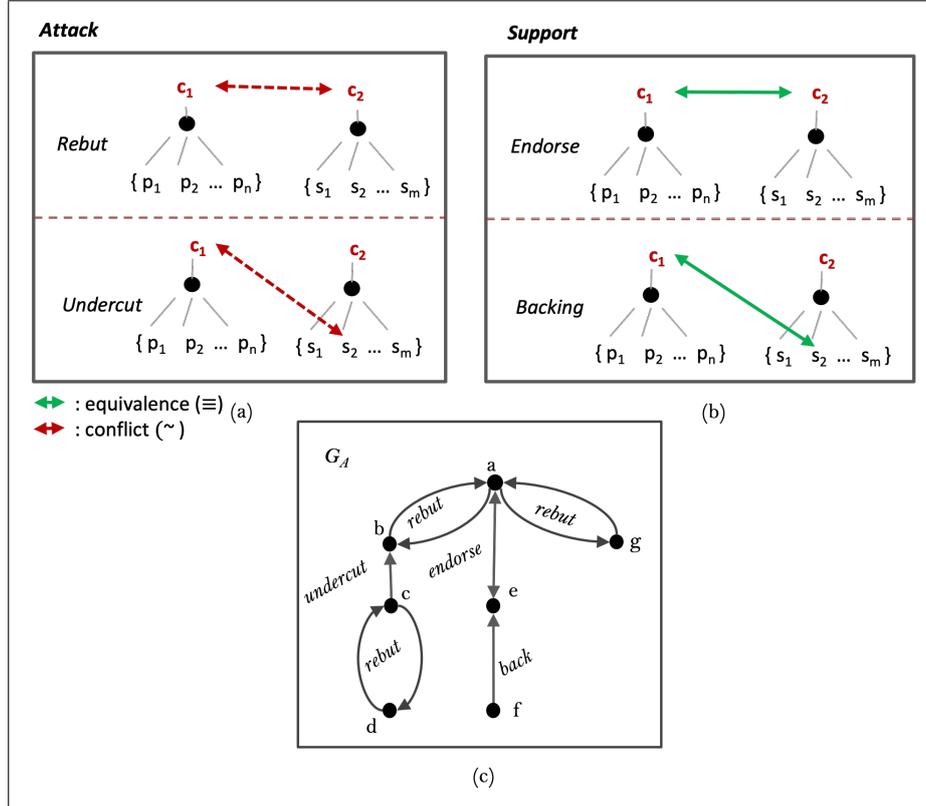


Figure 1: Example of data model

premise and conclusion part of arguments, respectively. More precisely, the second form of an argument pattern may be one of the following:

$$v_a : \langle \{p_1, \dots, p_n\}, c \rangle \quad \text{or} \quad v_a : \langle v_p[f], c \rangle$$

where  $p_1, \dots, p_n$  are constant propositional values,  $c$  is a proposition or variable,  $v_p$  is a variable and  $f$  a premise filter. Variable  $v_p$  matches the premise part of arguments and in particular, it matches sets of propositions, whereas  $c$  matches the conclusion. The occurrence of the expression  $[f]$  is optional. When it exists, the premise part is restricted based on a propositional set pattern, let  $s$ , which can either be a set of propositions (strings) or a variable that takes as value a set of propositions and, so, there can be 3 types of filters: *inclusion*, *join* and *disjointness*

written as  $[/s]$  ,  $[.s]$  and  $![s]$ , respectively. Below, are some examples of argument patterns:

- $\langle ?v[/\{ "p_1" \}], ?c \rangle$  : match arguments the premise of which include some proposition equivalent to "p<sub>1</sub>".
- $\langle ?v, "c" \rangle$  : match arguments with conclusion any proposition equivalent to "c"
- $\langle ?v[.\{ "p_1", "p_2" \}], ?c \rangle$  : match arguments whose premise intersects or is equivalent to the set  $\{ "p_1", "p_2" \}$
- $\langle \{ "p_1", "p_2" \}, "c" \rangle$  : instantiated arguments are also argument patterns

*Path patterns* are expressions that match complete paths and allow for navigation in the graph. They are identified by sequences of relations separated by the character '/' (e.g. `attack/support/support`). Note that a *relation* can either be one of the sub-relations (rebut, undercut, endorse, backing) or one of the general ones (attack, support). In the second case, any of the corresponding sub-relations will match the pattern. The expression  $*n$  is a syntactic sugar to express the " $n$  repetitions of a path pattern". For example `rebut*2`, is an alternative notation of `rebut/rebut`. In addition, we can express the case "up to  $n$  repetitions", by using the notation  $'+n'$ . In particular, `attack+3` defines three different patterns:  $\{ \text{attack}, \text{attack/attack}, \text{attack/attack/attack} \}$ . The existence of multiple '+' indicators in the same pattern defines a maximum number of combinations, equal to the proliferation of the number of '+'s. For example, the expression  $((\text{attack})+2)+3$  will give  $2 \times 3 = 6$  alternative path patterns: the '+2' indicator defines 2 path patterns, and for each of them the '+3' indicator will give 3 different patterns. Note the *path* clause in the *return* statement allows to return the whole parts of the graph that match the path patterns.

Next, we show some examples of complete queries in ArgQL:

Q1. Find arguments which are in a maximum distance of 3 "defend" (attack the attackers) relational steps, from arguments that have conclusion equivalent to "*Cloning includes ethical risks*", and return the complete matching paths.

```
match ?a1 (attack/attack)+3
?a2:<?pr, "Cloning includes ethical risks">
return path(?a1, ?a2)
```

Q2. Find and return arguments which attack other arguments connected via a sequence of three support relations, to an argument, for which, one of the premises is equivalent to the proposition *"cloning contributes positively in artificial insemination"*.

```
match ?arg attack/(support)*3
<?pr[/{"cloning contributes ... insemination"}],?c>
return ?arg
```

Q3. Find pairs of arguments whose premises intersect and return them.

```
match ?a1:<?pr1, ?c1>, ?a2:<?pr2[.?pr1], ?c2>
return ?a1, ?a2
```

### 3 Argument retrieval systems

In this section we explore tools for searching and retrieving argumentative data. In particular, we consider tools which move beyond a standard web search, where a user may be able to ask questions like “is coffee good for you?” and receive in return a number of articles discussing this topic. Instead we look at tools which provide the ability for more nuanced exploration of the arguments; such as “give me arguments pro and con why coffee is good for you” or “give me an argument from expert opinion supporting coffee being good for you”. The ability to search for arguments in this way has a wealth of potential applications, from assisting users in reaching decisions and forming opinions, to providing on-demand knowledge bases for dialogical agents.

Implementations of such argument search tools can be split into two broad categories: those for searching structured data, and those for searching unstructured data. In the former case, an interface is provided for searching an existing corpus of known argumentation structures gathered, for example, through manual annotation, guided argument construction, or as the record of a structured dialogue (see Section 4). In the latter case, searching in unstructured data, these tools generally combine a specific query with argument mining techniques to

determine the arguments contained in unstructured text. The argument mining approach allows for results to be found in a broader range of material that has not been previously analysed, though often with somewhat less accurate results. This categorisation corresponds to the two shared tasks introduced in Touché 2020 [Bondarenko *et al.*, 2020] which firstly looked at retrieval of arguments on socially important topics from a pre-existing set of structured arguments, and secondly, using argument mining techniques to retrieve documents with relevant arguments from a generic web crawl.

### 3.1 Tools for searching structured data

One of the first dedicated argument search engines to be developed was the *args.me* prototype [Wachsmuth *et al.*, 2017], which retrieves relevant arguments on a given query, ranks these according to their relevance, and presents them as lists of points pro and con (see Figure 2). In describing the development of *args.me* the authors noted that, at the time, no automatic argument mining approach seemed “robust enough, yet, to obtain arguments reliably from the web”. As such, *args.me* instead returns results using an index of structured arguments crawled from a number of online debate portals, specifically: *idebate.org*, *debatepedia.org*, *debatewise.org*, *debate.org*, and *forandagainst.com*. These portals all allow users to contribute pro and con arguments for given controversial issues, where the stance is explicitly indicated by the user, and therefore offers a source of data which can be reliably classified by this stance. In total 291,440 arguments were collected from these sources. While the *args.me* prototype only offers a simple interface for retrieving lists of pro and con arguments, it offers a framework for future expansion, improving the individual steps of retrieving, ranking and displaying arguments.

Of similar size to the *args.me* index is the Argument Interchange Format (AIF) database, AIFdb [Lawrence *et al.*, 2012b], which contains over 28,000 argument maps, with over 3.3m words and 270,000 claims in more than twenty different languages. However, being based on AIF, AIFdb contains a significantly richer and more fine-grained representation of argument structure, where individual propositions are connected by specific argumentative relationships (e.g. support or conflict) with these relationships able to be further specified as instances of a given

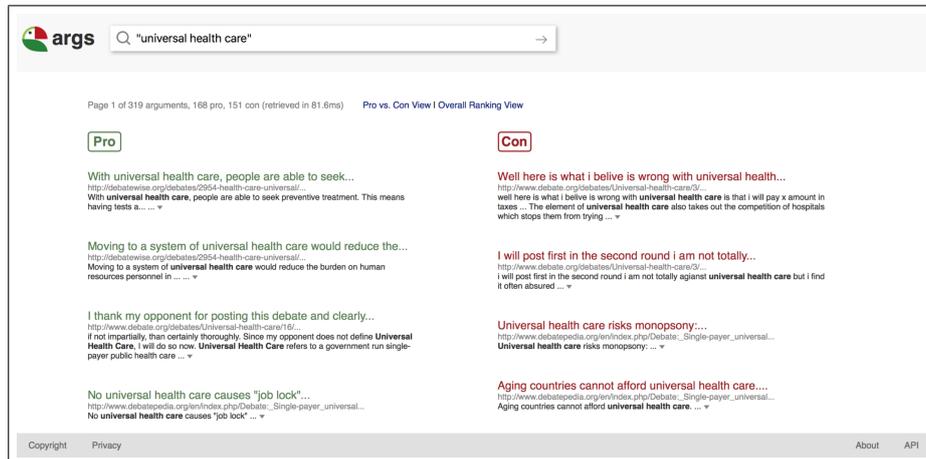


Figure 2: Args.me search results showing pro and con arguments on the topic of “universal health care”.

argumentation scheme. AIFdb offers three distinct search methods: a basic search functionality which matches the given text to propositions in the database; an advanced search functionality which offers the ability to narrow results by speaker, date, or argumentation scheme; and a web service interface allowing for direct queries to be performed on the underlying relational database. For the basic and advanced searches, results are returned as a list of matching elements with the ability to select any of these and view them in the context of the argument maps in which they appear (see Figure 3). For the web service interface, results are returned in the JavaScript Object Notation (JSON) format, allowing for processing by software tools. AIFdb’s native search interfaces do not perform ranking of the obtained results, however these have been combined with AIF *argument analytics* to give rankings by a number of measures including *Centrality*, which can be viewed as how important an issue is to the argument as a whole (calculated through eigenvector centrality, used in the Google Pagerank algorithm [Brin and Page, 1998]), and *Divisiveness*, which is used to assess how much an issue splits opinion (calculated based on how many other issues are in conflict with it and the amount of support which the two sides have) [Lawrence *et al.*, 2017].

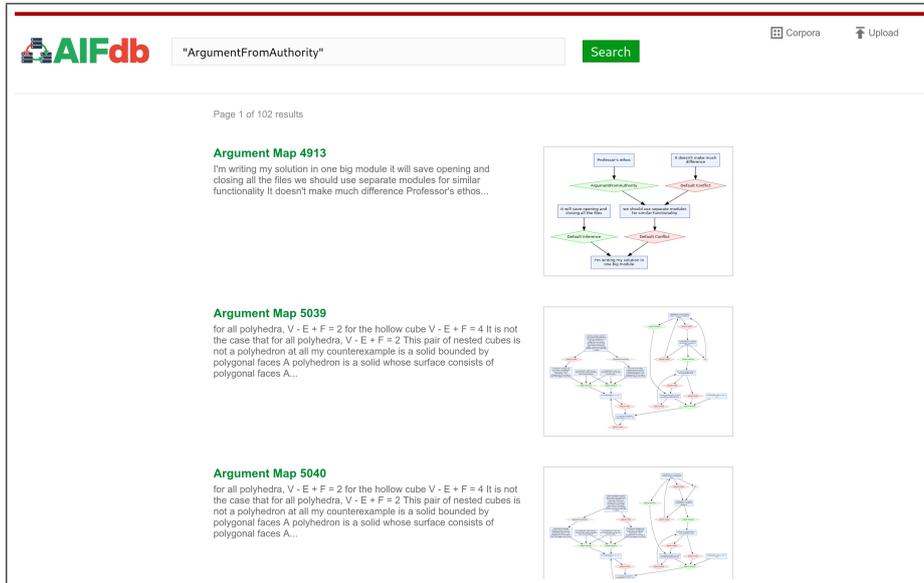


Figure 3: AIFdb advanced search results listing all matching instances of the *Argument from Authority* argumentation scheme.

In addition to those tools mentioned above, a number of online debate portals (e.g. CreateDebate<sup>1</sup>, Kialo<sup>2</sup> and PerspectroScope [Chen *et al.*, 2019]) as well as argument mapping tools (e.g. DebateGraph<sup>3</sup> or Rationale Online<sup>4</sup>) offer users some minimal search functionality, however in each case this is limited to text searches showing results for either individual claims, or an argument map containing the specified text, and does not allow for any more complicated querying of the argument structure.

### 3.2 Tools for searching unstructured data

Argument mining is the automatic identification and extraction of the structure of inference and reasoning expressed as arguments presented

<sup>1</sup><https://www.createdebate.com>

<sup>2</sup><https://www.kialo-edu.com>

<sup>3</sup><http://debategraph.org>

<sup>4</sup><https://www.rationaleonline.com/>

in natural language [Lawrence and Reed, 2020]. Whilst all applications of argument mining could be viewed as a form of argument search in that they find the argument structure contained within a given text, we constrain the discussion here to applications where either the argument components being mined match specific pre-determined search criteria, or where argument mining is applied in order to identify the full structure and some search technique is then used to return results based on this.

A prominent example in this first category can be found in much of IBM’s work on Project Debater<sup>5</sup>. Debater can respond to a given topic by automatically constructing a set of relevant pro/con arguments phrased in natural language. For example, when asked for responses to the topic “The sale of violent video games to minors should be banned”, an early prototype of Debater scanned approximately 4 million Wikipedia articles and determined the ten most relevant articles, scanned all 3,000 sentences in those articles, detected sentences which contain candidate claims, assessed their pro and con polarity and then presented three relevant pro and con arguments<sup>6</sup>, with more recent developments also working towards ranking and selecting the most convincing of these arguments [Gleize *et al.*, 2019], expanding the topic of the debate [Bar-Haim *et al.*, 2019], and providing “first principle” debate points, commonplace arguments which are relevant to many topics, where specific data is lacking [Bilu *et al.*, 2019]. In [Levy *et al.*, 2014], the challenge of searching for *Context Dependent Claims (CDCs)* in Wikipedia articles was first addressed, showing how, given a topic and a selection of relevant articles, a selection of “general, concise statements that directly support or contest the given topic” can be found. This work was followed in [Rinott *et al.*, 2015] where finding supporting evidence from Wikipedia data for a given CDC was addressed. [Bar-Haim *et al.*, 2017] introduced the task of claim stance classification, that is, detecting the target of a given CDC, and determining the stance towards that target. [Levy *et al.*, 2017] further developed CDC identification, removing the need for pre-selected relevant articles, by first deriving a *claim sentence query* to retrieve CDCs from a large unlabelled corpus. Such

---

<sup>5</sup><https://www.research.ibm.com/artificial-intelligence/project-debater/>

<sup>6</sup><http://www.kurzweilai.net/introducing-a-new-feature-of-ibms-watson-the-debater>

large volumes of CDCs can be used both as potential points to be made by the debater system as well as to aid in the interpretation of spoken material containing breaks, repetitions, or other irregularities [Lavee *et al.*, 2019].

Tools where argument mining is instead used as a component of argument search include ArgumenText [Stab *et al.*, 2018], an argument search system capable of retrieving pro and con arguments relevant to a given topic from the English part of the CommonCrawl Web corpus [Patel, 2020]. ArgumenText first retrieves a list of documents relevant to a given topic (where a topic is considered as “some matter of controversy that can be concisely expressed through keywords”) and then applies an argument mining model to identify the argument structure of the top-ranked documents; first classifying each document sentence as ‘argument’ or ‘no-argument’ with respect to the topic and then determining the stance (pro or con) of each topic-relevant argument. The results of a search on the topic of “self-driving cars” can be seen in Figure 4.

**ArgumenText** self-driving cars Search

Pro/Con List Weights Docs

Found 164 arguments (98 pro; 66 con) in 20 documents (classified 621 sentences in 2.921 ms)

**Filter by URL:**

- multivu.com (21)
- futurist.com (18)
- gizmodo.com (15)
- blog.cjponyparts.com (13)
- self-drivecar.com (11)
- ideas.time.com (10)
- businessinsider.com (9)
- bgr.com (9)
- thetruthaboutcars.com (7)
- extremetech.com (7)
- hybridcars.com (6)
- computerworld.com (6)
- dailybits.com (6)
- autoworldnews.com (6)
- whogotfunded.com (5)
- mobilenapps.com (5)
- slashgear.com (4)

**PRO:** Thanks to vehicle-to-vehicle and vehicle-to-infrastructure communication systems, autonomous cars and trucks could significantly reduce traffic congestion and traffic accidents. (0.9771)  
<http://www.futurist.com/2013/11/13/greener-future-self-driving-cars/>

**CON:** Because self-driving cars are powered by computers and computers can inevitably be hacked, there are some serious security concerns here. (0.9592)  
<http://gizmodo.com/whats-keeping-self-driving-cars-off-the-road-1450916024>

**PRO:** Self-driving vehicles can contribute to reducing infrastructure investments and enrich city life in other ways, such as by reducing emissions, and improving air quality and traffic safety. (0.9709)  
<http://www.multivu.com/mnr/84153-volvo-self-driving-cars-unique-swedish-project>

**CON:** However, it's important to point out that self-driving cars could also pose some unique safety problems of their own. (0.9545)  
<http://blog.cjponyparts.com/2014/08/are-we-road-self-driving-cars-info-graphic/>

**PRO:** Autonomous vehicles and a smarter infrastructure will bring us another step closer to even safer traffic and an improved environment. (0.9711)  
<http://www.multivu.com/mnr/84153-volvo-self-driving-cars-unique-swedish-project>

**CON:** For example, we haven't built self-driving cars to deal with intense conditions like busy city driving and extreme weather. (0.9204)  
<http://gizmodo.com/whats-keeping-self-driving-cars-off-the-road-1450916024>

**PRO:** This technology can also improve safety significantly, reduce fuel consumption and congestion. (0.9669)  
<http://www.multivu.com/mnr/84153-volvo-self-driving-cars-unique-swedish-project>

**CON:** The technology's not quite thereThe most obvious hurdle for self-driving cars is the technology that makes them drive themselves. (0.8659)  
<http://gizmodo.com/whats-keeping-self-driving-cars-off-the-road-1450916024>

Figure 4: Argumenttext search results showing arguments pro and con for the topic of “self-driving cars”.

Similarly, DebateLab<sup>7</sup> was an HFRI-funded project that intended to

<sup>7</sup><https://debatelab.ics.forth.gr/>

pioneer research towards developing the theoretical machinery that can be used across diverse domains for representing, mining and reasoning with online arguments [Ymeralli *et al.*, 2022]. It focused on journalistic articles written in the Greek language, and developed a suite of tools for article management<sup>8</sup>, most importantly a portal where users can view, search, visualise and analyse journalistic articles, the arguments they contain and their relationships. The articles and their arguments are automatically retrieved using a crawler and an argument mining module, while a Named Entity Recognition (NER) tool allows connecting entities mentioned in articles and arguments with online resources (e.g., wikipedia pages) that describe these entities. DebateLab supports various advanced searching functionalities, that employ, in the backend, the ArgQL language (see [Zografistou *et al.*, 2017], [Zografistou *et al.*, 2018], [Roussakis *et al.*, 2022] and also Section 2 in this Chapter), as well as advanced visualisations such as sunburst or tree-like views.

TARGER [Chernodub *et al.*, 2019] is an open source system combining argument mining techniques for tagging arguments in free text, with retrieval of arguments matching a given query. TARGER uses a pre-processing step to identify argumentative units and classify them as claims or premises. This pre-processing is carried out on the DepCC corpus [Panchenko *et al.*, 2018] to tag and store argument unit information in a web-scale index. The search component of TARGER allows the user to enter a keyword query and choose whether it should be matched in claims, premises, etc. Every retrieved result is rendered as a text fragment, with color-coded highlighting of each component’s role in the overall argument structure.

DISPUTool [Haddadan *et al.*, 2019], [Goffredo *et al.*, 2023] allows the exploration and identification of argumentative components over political debates, in an automated manner, using argument mining technologies. It has been applied to analyse political debates from the presidential campaigns of the USA since 1960, and is framed as a tool to support humanities’ scholars in exploring and analysing textual political debates. The argument mining component serves as a backend to a tool<sup>9</sup> that provides different functionalities on the analysed debates, in-

---

<sup>8</sup>[https://isl.ics.forth.gr/debatelab\\_portal/](https://isl.ics.forth.gr/debatelab_portal/)

<sup>9</sup><https://disputool.uni.lu/>, newer version in: <https://3ia-demos.inria.fr/>

cluding, among others, the (visual) exploration of claims, premises and their relationships, as well as the identification of named entities and fallacies.

## 4 Closed debating systems

In this section, we present several tools that have been used for debating, i.e., tools which allow the creation, management and/or analysis of debates. As with the rest of this chapter, the emphasis here is on the argument exploration and searching functionalities of the presented tools, many of which also allow various sophisticated visualisations and analytics. Our focus in this section lies on tools whose main functionality is the management and/or analysis of debates that have been created *within* the tool; this is in contrast to Section 3, which focused on systems that provide searching, analytics and exploration capabilities for arguments expressed elsewhere and searched by the tool, or retrieved by the tool for further analysis.

The term “debating” is often used in a very broad sense to include various types of textual interaction among users. Therefore, we decided to provide some criteria (requirements) as to which tools are candidates for inclusion in this section. These requirements are the following:

**Frontend** The tool should provide a frontend, allowing users to create, manage, search, explore and/or interact with arguments.

**Argument creation** The tool should provide functionalities allowing the expression of opinions or arguments, either as free text, or using some semi-structured format.

**Reactions to arguments** The tool should allow reactions (replies) to other people’s opinions in the form of new opinions that attack or support the original. We are not interested in other types of replies (e.g., responses to reviews that are used in standard reviewing tools, or the type of responses used in question-answering platforms like [stackoverflow](https://stackoverflow.com/)<sup>10</sup>). Votes and other types of structured reactions are possible, but not mandatory.

---

[r/disputool/](https://github.com/argdisputool/).

<sup>10</sup><https://stackoverflow.com/>

**Argument search and navigation** The tool should allow the user to search arguments, navigate through them and explore their contents. As this is the main focus of this chapter (and section), we prefer tools that provide advanced argument searching and navigation/exploration capabilities, including functionalities that allow analytics, advanced visualisations, and other sense-making features.

It should be noted that, even under this restrictive understanding, there are still numerous tools that fit this description and are relevant for this section. Thus, we had to make a selection, and only consider the most popular or significant ones. Importantly, some of the older systems that are presented here are no longer supported, and their URLs are non-functional; still, we decided to include them, on the basis of their importance in shaping the current state-of-the-art in the area. All the chosen systems are presented in Subsection 4.2 below. Before that, in Subsection 4.1, we present a number of concepts that are employed by the presented systems and may be useful for understanding their functionality.

#### 4.1 Debating models

In this section, we present three models for debating in the context of decision-making that are employed by some of the tools to be presented later (in Subsection 4.2).

The *Issue-Based Information Systems (IBIS)* model, proposed in [Kunz and Rittel, 1970], is employed by many of the tools below. IBIS is a model, based on argumentation, that is used to help users obtain a better understanding of so-called “wicked problems”, i.e., problems that are complex, ill-defined and involve multiple stakeholders. IBIS, as the name implies, is based on *issues*, which are questions that need to be answered by *positions*. Positions can be elaborated upon using *pro and con arguments*, i.e., arguments that support or object the respective position. Arguments themselves can be associated with other pro and con arguments that support or object the argument. Eventually, this creates an acyclic graph of arguments, rooted on an issue and including one or more positions and their pro/con arguments (and their pro/con

arguments recursively). This graphical representation can be used to support the dialectical process and to improve sense-making in wicked problems.

The *Quantitative Argumentation Debate* (*QuAD*) framework [Baroni *et al.*, 2013; Baroni *et al.*, 2015] can be seen as the formal computational counterpart of the IBIS model. A QuAD framework supports issues, positions and arguments, in the same manner as IBIS, the only exception being that, in QuAD, issues cannot be directly linked to arguments. Moreover, QuAD supports a numerical score assigned to each argument, which represents its strength, or importance, according to the domain experts. This score can be used for an automated quantitative evaluation of the debate, and the determination of the “winning” positions or arguments. For this, different evaluation methods have been proposed in the literature, including, indicatively, [Leite and Martins, 2011; Egilmez *et al.*, 2013; Patkos *et al.*, 2016b; Patkos *et al.*, 2016a] and others.

Decision matrices [Pugh, 1991] is another way to visualise wicked problems in order to make well-informed decisions. The model is based on a matrix, whose rows correspond to evaluation criteria, and whose columns correspond to different options. The possible options are evaluated against the criteria, resulting in a positive, negative or indifferent evaluation. The options are then assessed against the criteria, taking into account weights that represent the importance of each different criterion. Note that this allows an automated quantitative evaluation of the alternative options.

## 4.2 List of tools

In this subsection we present some of the most important debating tools that have appeared online or in the relevant literature. The tools are organised in groups based on their features, functionality and similarities. Recall that, according to our desiderata set forth in the beginning of this section, we are listing tools providing an appropriate *frontend* that supports *argument creation*, *reactions to arguments* and *argument search and navigation* capabilities.

**Tools based on the IBIS model.** Many of the tools that support debating are based on the IBIS model, or its computational counterpart, QuAD, described in Subsection 4.1. As a matter of fact, almost all of the academic tools that will be presented here are based on this model. Below, we present some of these tools.

APOPSIS<sup>11</sup> provides an IBIS-based platform to support decision-making dialogues [Ymeralli *et al.*, 2017]. APOPSIS allows users to enter arguments in a structured form, as well as to react, in different ways, to previously-submitted arguments. It proposes a two-phase discussion process: in the first phase, all possible ideas for solutions (called *positions*) to the *issue* under discussion are gathered, and the most promising of them (based on users’ reactions) are selected; in the second phase, no further positions can be introduced, and discussion focuses on those selected positions, in order to understand better their pros and cons. The process is supported by an automated argument evaluation procedure, based on sm-dice [Patkos *et al.*, 2016a; Patkos *et al.*, 2016b], as well as by sophisticated analytics and visualisations for identifying trends and patterns related to users, their characteristics, and their expressed positions and arguments.

DesignVUE [Auricchio and Bracewell, 2013; Baroni *et al.*, 2013; Baroni *et al.*, 2015] was a decision-support system based on the IBIS and QuAD models. It supported the creation of IBIS debates and their automated evaluation. Unfortunately, its online version is no longer available.

A very similar tool developed by the same group was Quaestio-it [Evripidou and Toni, 2014], which used to be a popular system for creating, browsing, analysing and visualising arguments. The website of quaestio-it<sup>12</sup> is no longer maintained, and, to the best of our knowledge, the tool is no longer available. Quaestio-it allowed both attacking and supporting arguments.

Arg&Dec [Auricchio *et al.*, 2015] (standing for “Argue & Decide”) was the third tool by the same group (developed after DesignVUE and Quaestio-it), with similar, but more advanced, functionalities. Unlike

---

<sup>11</sup><https://demos.isl.ics.forth.gr/apopsis/>

<sup>12</sup>[www.quaestion-it.com](http://www.quaestion-it.com)

the other two, it is still supported and available as a web application<sup>13</sup>. Arg&Dec aims to assist collaborative decision-making through debating and argumentation. It is free to use (sign-up required), and supports collaborative work. Emphasis is placed on the mode of interaction (and decision-making) and two modes are supported, one based on QuAD frameworks and one based on Decision Matrices (see Subsection 4.1 for details on those decision-making models). Under the QuAD mode of interaction, users can create and edit different node types and (pro/con) links between them, as provisioned by the QuAD model. Under the Decision Matrices model, the users can create the matrix's rows and columns (evaluation criteria and options respectively), edit the weights of rows and the generated cells. The two models are interchangeable, in the sense that a Decision Matrix can be transformed to a QuAD and vice-versa. Under both modes, the system provides a visual representation of the input, and is able to automatically evaluate the debate and propose prevailing options, along with a numerical score. To further support the decision-maker, natural language explanations regarding option ranking are provided, including a speech synthesis functionality.

COLLAGREE [Ito *et al.*, 2015] is an open web-based forum system aiming to allow large-scale discussion for the purposes of agreeing on a certain matter (consensus-building). The system employs facilitators, whose aim is to coordinate, lead, integrate, classify, and summarise discussions towards consensus. Emphasis is placed on helping facilitators moderate discussions among many people. The discussions are based on published issues (in the sense of IBIS), but no strict structure on the discussion is imposed, and people can submit their opinions as plain text. Gamification techniques are employed to encourage participation. Keyword-based search facilities are provided, as well as some functionalities for sense-making (e.g., sentiment analysis, automated keyword extraction) and simple visualisations. The system has been employed in large-scale field experiments, e.g., for an internet-based town meeting in Nagoya, Japan [Ito *et al.*, 2014] and in the Aichi prefecture, Japan [Ito, 2021].

D-Agree [Ito *et al.*, 2019], [Ito *et al.*, 2022], [Ito, 2021] is a system very similar to COLLAGREE, and can be viewed as its continuation in

---

<sup>13</sup><http://arganddec.com/>

many aspects. Just like COLAGREE, D-Agree uses the IBIS discussion model, gamification techniques, keyword-based searching facilities and visualisations based on a tag cloud. However, D-Agree adds a critical component compared to COLLAGREE: acknowledging the difficulty of the moderation task in large-scale discussions, D-Agree employs an automated facilitation agent that uses deep learning and natural language processing techniques to support the discussion. In particular, the facilitator agent is used to capture meaningful sentences and extract the discussion structure from the texts posted by users in discussions (i.e., identifying issues, positions and arguments in the posted free text), so that users' interaction is more natural, while at the same time adhering to the IBIS model. Importantly, the automated facilitator also prompts the users for additional information, keeping the discussion alive. D-Agree was also tested in a field test experiment in Nagoya, Japan, as well as in small-scale controlled experiments. The authors' experimental results show that the use of the automated facilitator agent led to a more lively discussion compared to the scenario of using human facilitators.

Deliberatorium<sup>14</sup> is a rather old, but still active web-based system for enabling people have productive discussions about complex (so-called wicked) problems [Gurkan *et al.*, 2010]. As in other tools in this subsection, Deliberatorium follows the IBIS model. It organises contributions by topic, rather than submission time, using argumentation maps, and ensures that cycles and repetitions are kept to a minimum. Towards this, authors of arguments and other contributions need to follow certain rules, while moderators are employed to ensure that these rules are adhered to. In terms of argument search, the functionalities are really basic: some standard keyword-based searching facilities are supported, and the discussions are organised in a tree-based structure inspired by the IBIS model, facilitating sense-making.

**Other online debating platforms.** Various online debating platforms have emerged from a non-academic setting (e.g., Kialo, CreateDebate). These platforms aim to allow online communities to engage in discussions and express their opinions. Their business models vary, as will be seen below.

---

<sup>14</sup><https://deliberatorium.mit.edu/>

Kialo<sup>15</sup> is a website run by Kialo Inc., with the aim to “cut through the noise typically associated with social and online media, making it easy to engage in focused discussion”. It supports the creation of and participation in debates on any conceivable topic. It provides numerous features for organising the ensuing dialogues and the involved arguments, allowing easy sense-making of lengthy dialogues, as well as different visualisations of the debate in the form of an argument tree or sunburst diagram. After login, users can create their own arguments (or debates), as well as respond and/or react to other users’ arguments in different ways. Importantly, Kialo allows claims to be associated with external sources (e.g., scientific reports or articles) that back them up. A custom version of this platform for educators<sup>16</sup> has been used for class discussion and critical thinking and reasoning, teaching logical fallacies etc. Kialo is a very interesting success story, as it contains thousands of debates and arguments. The website is free to use, and does not display advertisements or sell users’ data. Therefore, it produces no revenue for its maintainer. Instead, the website is used as a “demonstrator” of Kialo’s capabilities, aiming to sell those functionalities to companies as a deliberation and decision-making tool.

CreateDebate<sup>17</sup> is an online platform that allows users to post a dilemma, in the form of a statement, and to provoke a debate over the truth of this statement. The debates consist of textual arguments that people post as a response to the original statement. An argument can be further supported or disputed by other users using their own arguments, and users can also ask for clarifications over an argument. The debate is organised into pro and con “sides”, and the system automatically computes a score for each “side”, encouraging participation in order to “win” the argument. It also provides some limited forms of visualisation allowing users to better understand the flow of the debate. The system’s revenue model seems to be ad-based, and, in fact, the system is often being misused for ad-related postings that are irrelevant to its stated purpose. Nevertheless, it constitutes one of the first systems that attempted to create a global debating forum of users, aiming at a

---

<sup>15</sup><https://www.kialo.com/>

<sup>16</sup><https://www.kialo-edu.com>

<sup>17</sup><https://www.createdebate.com/>

providing a podium for publicly expressing one’s opinion on important matters.

A similar tool allowing online debates is the web-based discussion software *Arvina* [Lawrence *et al.*, 2012a]. Unlike the tools mentioned earlier in this category, *Arvina* emerged from academic work. *Arvina* allows participants to debate a range of topics in real-time in a way that is structured but at the same time unobtrusive. *Arvina* maintains flexibility in dialogue structure by using protocols written using the Dialogue Game Description Language (DGDL) [Bex *et al.*, 2014] to structure the discussion between participants. Such protocols determine which types of moves can be made (e.g. questioning, claiming, etc.), when these moves can be made (e.g. a dialogue starts with a claim; question moves can only be made in the turn directly following a claim; etc.), and describe how each move updates the argument structure of the discussion taking place.

*Arvina* can support multiple human users interacting in the same dialogue, as well as incorporating software agents representing (the arguments of) specific authors who have their opinions stored in AIFdb [Lawrence *et al.*, 2012b]. So, for example, say that a user has constructed a complex, multi-layered argument using the OVA argument analysis tool [Janier *et al.*, 2014], concerning the use of nuclear weapons. An agent representing this user can then be added to an *Arvina* discussion and questioned about these opinions, with the agent answering by giving the user’s pre-annotated opinions.

**Tools employing structured arguments.** All the debating tools presented above support free-text, unstructured arguments. Thus, the structure they impose lies on the debate level, rather than the argument level. Two debating tools, Gorgias-B [Spanoudakis *et al.*, 2021; Spanoudakis *et al.*, 2022], and Carneades [Walton and Gordon, 2012], [Walton and Gordon, 2017] stand out from the rest in the sense that they employ structured arguments, which allow them to provide more sophisticated analysis and reasoning functionalities. We present them below.

Gorgias-B [Spanoudakis *et al.*, 2021; Spanoudakis *et al.*, 2022] is a

reasoning and argumentation tool<sup>18</sup>, developed through a series of publications, and based on an earlier tool called Gorgias<sup>19</sup>. The stated purpose of Gorgias-B is to allow users to make informed decisions over a set of options, based on a thorough modelling of the underlying situation. Gorgias-B does not take as input free-text arguments; instead, it prompts registered users to define facts (predicates) and to connect them to form structured arguments (logical rules) and counter-arguments (or exceptions). Preferences, priorities and default options can also be defined graphically. In this sense, it can be seen as a no-code programming tool for declarative programming, with argumentation semantics. Once the domain has been modelled, specific scenarios (i.e., situations in which certain facts are true and others are false) can be input to the system in order for that scenario to be evaluated (reasoned upon) automatically by the system based on the domain knowledge. Explanations over the system's output are also provided.

Carneades [Walton and Gordon, 2012], [Walton and Gordon, 2017] is a tool<sup>20</sup> for representing and evaluating arguments, currently in its fourth version. Carneades uses a graph-based representation of arguments, where each node of the graph can be either a statement or an argument. Statements represent propositions that can be true or false, and function as premises or conclusions of arguments. The premises and conclusions of a given argument are determined via links (edges) between the respective argument node and the statements that represent its premises and conclusions, thereby explicitly representing the structure of an argument. Carneades is heavily based on previous work by Walton on argumentation schemes [Macagno *et al.*, 2018], which are used for reasoning, argument evaluation, as well as for argument invention, i.e., the construction of arguments from statements known to be true. In terms of interaction with the user, Carneades supports a visual display of arguments, which is based on the user's specification of the statements considered true and on the ensuing evaluation which determines the (non-)acceptable arguments.

---

<sup>18</sup><http://gorgiasb.tuc.gr/>

<sup>19</sup><http://www.cs.ucy.ac.cy/~nkd/gorgias/>

<sup>20</sup><https://carneades.github.io/>

## 5 Conclusion

The evolution of the Web from a unidirectional information conduit to an interactive platform for global discourse has redefined the nature of user engagement, with the capacity for active contribution, debate, and exchange of ideas permeating every facet of online communication. Arguments, discussions, and viewpoints proliferate across diverse platforms, from social media to specialised forums, embodying a rich collection of perspectives and insights. Yet, amidst this expansive realm of user-generated content lies the challenge of navigating and accessing pertinent arguments efficiently.

As the volume of opinions and arguments grows, the significance of tools for discerning relevant content becomes paramount. Traditional keyword-based searches, while helpful, fall short when confronted with the intricacies of argumentative content. Arguments possess inherent structure and interconnectedness, demanding nuanced search capabilities beyond simplistic keyword queries. The quest for suitable tools intertwines with the necessity to encompass the structural complexities inherent in debates, prompting the exploration of specific tools for creating, navigating and searching online debates. In this chapter we have explored three fundamental areas: structured argument query languages, argument retrieval systems, and closed debating systems. Although substantial advances have been made in each of these, it is evident that there is still much to be done to improve their utility and reach.

### 5.1 Future directions

Existing argument search tools are often limited in both their scope and functionality, relying on pre-structured material and limiting the results returned to either related documents, or lists of reasons pro and con. Advances in Argument Mining may help to address the first of these issues, opening up unstructured text to the same search techniques currently available for structured argument data. In parallel to broadening the scope of data which can be searched, improvements can also be made in the types of search available, from simple keyword search identifying related arguments, to the ability for a user to provide a (partially specified) argument graph as input and return arguments whose structure

corresponds to the input graph.

Argument search can be broken down into a number of individual component steps: retrieval of relevant documents, identification of target argument components within these documents, ranking of components, and display of the identified results. While the first of these steps, retrieval of relevant documents, is an essential task, it largely relies on improvements in standard information retrieval techniques, and does not strictly depend upon any argument structure. The remaining three tasks on the other hand are all areas in which argument search tools could be directly improved. In terms of identifying target argument components within documents, not only can accuracy be improved by ongoing advances in argument mining, but as these techniques mature, the possibilities for more detailed queries grow. For example, a user could specify a particular graph as input and the search tool could return arguments whose structure corresponds to the input graph. Similar advances are also possible in improving the ranking of results. For example, incorporating elements of Argument Analytics [Lawrence *et al.*, 2016] to supplement relevance with factors such as criticality or divisiveness of returned arguments. Finally, although some search tools offer visualisations of results, many just display lists of pro and con points. In these cases, there is considerable potential for incorporating richer visualisations which place arguments in their broader context and allow for fluid exploration of related points.

Whilst current debating systems allow for creation, management and analysis of debates, these tools almost exclusively represent isolated platforms lacking in interoperability and wider adoption. Future work in this area needs to address this issue, developing shared underlying concepts and a unified data fabric, as well as providing opportunities for integration with existing online platforms. The ability for a user to highlight a span of text anywhere on the Web, see reasons for and against from a broad range of sources, and provide their own structured response, would move such tools beyond niche communities and open them up to a global audience.

## References

- [Auricchio and Bracewell, 2013] Marco Auricchio and Robert Bracewell. Capturing an integrated design information space with a diagram-based approach. *Journal of Engineering Design*, 24(6):397–428, 2013.
- [Auricchio et al., 2015] Marco Auricchio, Pietro Baroni, Dario Pellegrini, and Francesca Toni. Comparing and integrating argumentation-based with matrix-based decision support in arg&dec. In Elizabeth Black, Sanjay Modgil, and Nir Oren, editors, *Theory and Applications of Formal Argumentation*, pages 1–20, Cham, 2015. Springer International Publishing.
- [Bar-Haim et al., 2017] Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. Stance classification of context-dependent claims. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 251–261, Valencia, Spain, 2017.
- [Bar-Haim et al., 2019] Roy Bar-Haim, Dalia Krieger, Orith Toledo-Ronen, Lilach Edelstein, Yonatan Bilu, Alon Halfon, Yoav Katz, Amir Menczel, Ranit Aharonov, and Noam Slonim. From surrogacy to adoption; from bitcoin to cryptocurrency: Debate topic expansion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 977–990, Florence, Italy, 2019. Association for Computational Linguistics.
- [Baroni et al., 2013] P. Baroni, M. Romano, F. Toni, M. Auricchio, and G. Bertanza. An argumentation-based approach for automatic evaluation of design debates. In *CLIMA XIV: Proceedings of the 14th International Workshop on Computational Logic in Multi-Agent Systems - Volume 8143*, pages 340–256, 2013.
- [Baroni et al., 2015] P. Baroni, M. Romano, F. Toni, M. Auricchio, and G. Bertanza. Automatic evaluation of design alternatives with quantitative argumentation. *Argument and Computation*, 6:24–29, 2015.
- [Bex et al., 2014] Floris Bex, John Lawrence, and Chris Reed. Generalising argument dialogue with the dialogue game execution platform. In S. Parsons, N. Oren, C. Reed, and F. Cerutti, editors, *Proceedings of the Fifth International Conference on Computational Models of Argument (COMMA 2014)*, pages 141–152, Pitlochry, Scotland, 2014. IOS Press.
- [Bilu et al., 2019] Yonatan Bilu, Ariel Gera, Daniel Hershovich, Benjamin Sznajder, Dan Lahav, Guy Moshkovich, Anael Malet, Assaf Gavron, and Noam Slonim. Argument invention from first principles. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1013–1026, Florence, Italy, 2019. Association for Computational Linguistics.

- [Bondarenko *et al.*, 2020] Alexander Bondarenko, Maik Fröbe, Meriem Beloucif, Lukas Gienapp, Yamen Ajjour, Alexander Panchenko, Chris Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. Overview of Touché 2020: Argument Retrieval. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction. 11th International Conference of the CLEF Association (CLEF 2020)*, volume 12260 of *Lecture Notes in Computer Science*, pages 384–395, Berlin Heidelberg New York, September 2020. Springer.
- [Brin and Page, 1998] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.
- [Cabrio *et al.*, 2013] Elena Cabrio, Serena Villata, and Fabien Gandon. A support framework for argumentative discussions management in the web. In *The Semantic Web: Semantics and Big Data: 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings 10*, pages 412–426. Springer, 2013.
- [Chen *et al.*, 2019] Sihao Chen, Daniel Khashabi, Chris Callison-Burch, and Dan Roth. PerspectroScope: A window to the world of diverse perspectives. In Marta R. Costa-jussà and Enrique Alfonseca, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 129–134, Florence, Italy, July 2019. Association for Computational Linguistics.
- [Chernodub *et al.*, 2019] Artem Chernodub, Oleksiy Oliynyk, Philipp Heidenreich, Alexander Bondarenko, Matthias Hagen, Chris Biemann, and Alexander Panchenko. Targer: Neural argument mining at your fingertips. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 195–200, 2019.
- [Chesñevar *et al.*, 2006] Carlos Chesñevar, Jarred McGinnis, Sanjay Modgil, Iyad Rahwan, Chris Reed, Guillermo Simari, Matthew South, Gerard Vreeswijk, and Steven Willmott. Towards an Argument Interchange Format. *The Knowledge Engineering Review*, 21(04):293–316, 2006.
- [Cimiano *et al.*, 2013] Philipp Cimiano, Oscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph. *The Semantic Web: Semantics and Big Data: 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings*, volume 7882. Springer, 2013.
- [Dung, 1995] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.
- [Egilmez *et al.*, 2013] Sinan Egilmez, João G. Martins, and João Leite. Extending social abstract argumentation with votes on attacks. In Elizabeth

Black, Sanjay Modgil, and Nir Oren, editors, *Theory and Applications of Formal Argumentation - Second International Workshop, TFAFA 2013, Beijing, China, August 3-5, 2013, Revised Selected papers*, volume 8306 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2013.

- [Evrpidou and Toni, 2014] Valentinos Evripidou and Francesca Toni. Quaestio-it.com: a social intelligent debating platform. *Journal of Decision Systems*, 23(3):333–349, 2014.
- [Gleize *et al.*, 2019] Martin Gleize, Eyal Shnarch, Leshem Choshen, Lena Dankin, Guy Moshkovich, Ranit Aharonov, and Noam Slonim. Are you convinced? choosing the more convincing evidence with a Siamese network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 967–976, Florence, Italy, 2019. Association for Computational Linguistics.
- [Goffredo *et al.*, 2023] Pierpaolo Goffredo, Elena Cabrio, Serena Villata, Shohreh Haddadan, and Jhonatan Sanchez. Disputool 2.0: A modular architecture for multi-layer argumentative analysis of political debates. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37:16431–16433, 06 2023.
- [Gurkan *et al.*, 2010] Ali Gurkan, Luca Iandoli, Mark Klein, and Giuseppe Zollo. Mediating debate through on-line large-scale argumentation: Evidence from the field. *Information Sciences*, 180:3686–3702, 2010.
- [Haddadan *et al.*, 2019] Shohreh Haddadan, Elena Cabrio, and Serena Villata. Disputool – a tool for the argumentative analysis of political debates. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6524–6526. International Joint Conferences on Artificial Intelligence Organization, 7 2019.
- [Ito *et al.*, 2014] Takayuki Ito, Yuma Imi, Takanori Ito, and Eizo Hideshima. COLLAGREE: Facilitator-mediated large-scale consensus support system. In *Proceedings of the ACM Collective Intelligence Conference Series*, 2014.
- [Ito *et al.*, 2015] Takayuki Ito, M. Okumura, and E. Hideshima. Implementation of a large-scale discussion support system collagree - large-scale discussion support based on a weakly structured discussion process. *Journal of Japan Industrial Management Association*, 66:83–108, 01 2015.
- [Ito *et al.*, 2019] T. Ito, D. Shibata, S. Suzuki, N. Yamaguchi, T. Nishida, K. Hiraishi, and K. Yoshino. Agent that facilitates crowd discussion. In *Proceedings of the ACM Collective Intelligence Conference Series*, 2019.
- [Ito *et al.*, 2022] Takayuki Ito, Rafik Hadfi, and Shota Suzuki. An agent that facilitates crowd discussion. *Group Decision and Negotiation*, 31:621–647, 2022.
- [Ito, 2021] Takayuki Ito. *Discussion and Negotiation Support for Crowd-Scale*

- Consensus*, pages 371–393. Springer International Publishing, Cham, 2021.
- [Janier *et al.*, 2014] Mathilde Janier, John Lawrence, and Chris Reed. OVA+: An argument analysis interface. In S. Parsons, N. Oren, C. Reed, and F. Cerutti, editors, *Proceedings of the Fifth International Conference on Computational Models of Argument (COMMA 2014)*, pages 463–464, Pitlochry, Scotland, 2014. IOS Press.
- [Kunz and Rittel, 1970] W. Kunz and H. Rittel. *Issues as elements of information systems. Working Paper 131*. Institute of Urban and Regional Development, University of California, Berkeley, California, 1970.
- [Lavee *et al.*, 2019] Tamar Lavee, Matan Orbach, Lili Kotlerman, Yoav Kantor, Shai Gretz, Lena Dankin, Michal Jacovi, Yonatan Bilu, Ranit Aharonov, and Noam Slonim. Towards effective rebuttal: Listening comprehension using corpus-wide claim mining. In *Proceedings of the 6th Workshop on Argument Mining*, pages 58–66, Florence, Italy, 2019. Association for Computational Linguistics.
- [Lawrence and Reed, 2020] John Lawrence and Chris Reed. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818, 2020.
- [Lawrence *et al.*, 2012a] John Lawrence, Floris Bex, and Chris Reed. Dialogues on the argument web: Mixed initiative argumentation with arvina. In *Proceedings of the 4th International Conference on Computational Models of Argument (COMMA 2012)*, pages 513–514, Vienna, Austria, 2012. IOS Press.
- [Lawrence *et al.*, 2012b] John Lawrence, Floris Bex, Chris Reed, and Mark Snaith. AIFdb: Infrastructure for the argument web. In *Proceedings of the Fourth International Conference on Computational Models of Argument (COMMA 2012)*, pages 515–516, Vienna, Austria, 2012. IOS Press.
- [Lawrence *et al.*, 2016] J. Lawrence, R. Duthie, K. Budzysnka, and C.A. Reed. Argument analytics. In P. Baroni, M. Stede, and T. Gordon, editors, *Proceedings of the Sixth International Conference on Computational Models of Argument (COMMA 2016)*, Berlin, 2016. IOS Press.
- [Lawrence *et al.*, 2017] John Lawrence, Mark Snaith, Barbara Konat, Katarzyna Budzysnka, and Chris Reed. Debating technology for dialogical argument: Sensemaking, engagement, and analytics. *ACM Transactions on Internet Technology*, 17(3):24:1–24:23, 2017.
- [Leite and Martins, 2011] João Leite and João G. Martins. Social abstract argumentation. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 2287–2292. IJCAI/AAAI, 2011.
- [Levy *et al.*, 2014] Ran Levy, Yonatan Bilu, Daniel Hershovich, Ehud Aharoni, and Noam Slonim. Context dependent claim detection. In *Proceedings of the 25th International Conference on Computational Linguistics*, pages

1489–1500, Dublin, Ireland, 2014.

- [Levy *et al.*, 2017] Ran Levy, Shai Gretz, Benjamin Sznajder, Shay Hummel, Ranit Aharonov, and Noam Slonim. Unsupervised corpus-wide claim detection. In *Proceedings of the 4th Workshop on Argument Mining*, pages 79–84, Copenhagen, Denmark, 2017. Association for Computational Linguistics.
- [Macagno *et al.*, 2018] Fabrizio Macagno, Douglas Walton, and Chris Reed. *Argumentation Schemes*, pages 517–574. 08 2018.
- [Panchenko *et al.*, 2018] Alexander Panchenko, Eugen Ruppert, Stefano Falli, Simone P. Ponzetto, and Chris Biemann. Building a web-scale dependency-parsed corpus from CommonCrawl. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA).
- [Patel, 2020] Jay M. Patel. *Introduction to Common Crawl Datasets*, pages 277–324. Apress, Berkeley, CA, 2020.
- [Patkos *et al.*, 2016a] Theodore Patkos, Antonis Bikakis, and Giorgos Flouris. A multi-aspect evaluation framework for comments on the social web. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR-16), Short Paper*, 2016.
- [Patkos *et al.*, 2016b] Theodore Patkos, Giorgos Flouris, and Antonis Bikakis. Symmetric multi-aspect evaluation of comments (extended abstract). In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI-16), Short Paper*, 2016.
- [Pugh, 1991] S. Pugh. *Total Design: Integrated Methods for Successful Product Engineering*. Addison-Wesley, Wokingham, 1991.
- [Rinott *et al.*, 2015] Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M Khapra, Ehud Aharoni, and Noam Slonim. Show me your evidence—an automatic method for context dependent evidence detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 440–450, Lisbon, Portugal, 2015.
- [Roussakis *et al.*, 2022] Yannis Roussakis, Giorgos Flouris, Dimitra Zografistou, and Elisjana Ymeralli. Extending the ArgQL specification. In *Proceedings of the 6th International Joint Conference on Rules and Reasoning (RuleML+RR-22), 16th International Rule Challenge session*, 2022.
- [Spanoudakis *et al.*, 2021] N. Spanoudakis, K. Kostis, and K. Mania. "web-gorgias-b": Argumentation for all. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART 2021)*, 2021.
- [Spanoudakis *et al.*, 2022] N. Spanoudakis, G. Gligoris, A.C. Kakas, and A. Koumi. Gorgias cloud: On-line explainable argumentation (system

- demonstration). In *9th International Conference on Computational Models of Argument (COMMA 2022)*, 2022.
- [Stab *et al.*, 2018] Christian Stab, Johannes Daxenberger, Chris Stahlhut, Tristan Miller, Benjamin Schiller, Christopher Tauchmann, Steffen Eger, and Iryna Gurevych. Argumentext: Searching for arguments in heterogeneous sources. In *Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: demonstrations*, pages 21–25, 2018.
- [Wachsmuth *et al.*, 2017] Henning Wachsmuth, Martin Potthast, Khalid Al Khatib, Yamen Ajjour, Jana Puschmann, Jiani Qu, Jonas Dorsch, Viorel Morari, Janek Bevendorff, and Benno Stein. Building an argument search engine for the web. In *Proceedings of the 4th Workshop on Argument Mining*, pages 49–59, 2017.
- [Walton and Gordon, 2012] Douglas Walton and Thomas F. Gordon. The carneades model of argument invention. *Pragmatics & Cognition*, 20:1–31, 2012.
- [Walton and Gordon, 2017] Douglas Walton and Thomas F. Gordon. Argument invention with the carneades argumentation system. *SCRIPTed*, 14, 2017.
- [Ymeralli *et al.*, 2017] Elisjana Ymeralli, Giorgos Flouris, Theodore Patkos, and Dimitris Plexousakis. APOPSIS: A web-based platform for the analysis of structured dialogues. In *Proceedings of the 16th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE-17)*, 2017.
- [Ymeralli *et al.*, 2022] Elisjana Ymeralli, Giorgos Flouris, Vasilis Efthymiou, Katerina Papantoniou, Theodore Patkos, Georgios Petasis, Nikiforos Pittaras, Giannis Roussakis, and Elias Tzortzakakis. Representing online debates in the context of e-journalism. In *Proceedings of the 16th International Conference on Advances in Semantic Processing (SEMAPRO-22), Best Paper Award*, 2022.
- [Zografistou *et al.*, 2017] Dimitra Zografistou, Giorgos Flouris, and Dimitris Plexousakis. ArgQL: A declarative language for querying argumentative dialogues. In *Proceedings of the International Joint Conference on Rules and Reasoning 2017 (RuleML+RR), as the Best Paper of the Doctoral Consortium*, 2017.
- [Zografistou *et al.*, 2018] Dimitra Zografistou, Giorgos Flouris, Theodore Patkos, and Dimitris Plexousakis. Implementing the ArgQL query language. In *Proceedings of the 7th International Conference on Computational Models of Argument (COMMA-18), short paper*, 2018.