An Argument-based Search Framework: Implementation on a Spanish Corpus in the E-Participation Domain

Andrés Segura-Tinoco[1], Óscar G. Borzdynski[1] and Iván Cantador[1,*]

[1]*Escuela Politécnica Superior, Universidad Autónoma de Madrid, Spain*

Abstract

There are many domains and applications where generated textual content is rich with argumentative information, such as product reviews, online forum discussions, court orders, and parliamentary debates. In all of them, the automatic extraction and search of arguments can be very valuable for decision and policy making purposes, and represent challenging problems. Aiming to address these problems, we propose a general and flexible information retrieval framework which, in addition to text documents relevant for a given query, returns categorized and linked argumentative structures existing in or related to such documents within a collection. The framework is composed of a pipeline of modules targeting several tasks: text processing, argument-based annotation, argument mining, information retrieval, reranking and evaluation. As a proof of concept, we have implemented and tested the framework on a Spanish corpus with citizen proposals and comment threads from an e-participation platform.

Keywords

argument retrieval, argument mining, citizen participation

1. Introduction

Nowadays, there is a plethora of interactive technologies and digital channels that promote the generation of textual content rich with argumentative information. Examples of these systems are social media (e.g., online social networks, blogs, and microblogging services) where people express opinions and explain the reasons in favor or against such opinions, e-commerce sites where costumers provide detailed reviews about pros and cons of products, and web forums where users discuss a variety of topics. Besides, there are many domains and applications in which electronic documents record transcriptions of argumentative discourses. They include court orders in law, parliamentary debates in politics, and proposals in citizen participation, to name a few.

In all these cases, the automatic extraction and search of arguments and the relations between them can be very valuable to support decision and policy making. However, they represent

CIRCLE'22: 2nd Joint Conference of the Information Retrieval Communities in Europe, July 4–7, 2022, Samatan, Gers, France

*Corresponding author.

andres.segurat@uam.es (A. Segura-Tinoco); oscar.gomezb@estudiante.uam.es (: G. Borzdynski); ivan.cantador@uam.es (I. Cantador)
https://ansegura7.github.io/ (A. Segura-Tinoco); http://www.eps.uam.es/~cantador/ (I. Cantador)
0000-0001-6868-1445 (A. Segura-Tinoco); 0000-0001-6663-4231 (I. Cantador)

CEUR Workshop Proceedings (CEUR-WS.org)

challenging problems that entail a number of complex tasks [1]. They first require the formulation of an argument model, including argument components (e.g., *claims* and *premises*) and relations (e.g., *support* and *attack*). They require the development of computational methods to identify, delimit and classify those elements in input texts. Finally, they need specific information retrieval, summarization and visualization approaches. Moreover, most of these tasks also involve the creation and use of argument-based annotation corpora.

Within the information retrieval field, *argument search* (a.k.a. *argument retrieval*) is gaining momentum, as evidenced by the organization of the Touché Argument Retrieval labs [2, 3] at the 2020-2022 editions of the Conference and Labs of the Evaluation Forum (CLEF). So far, in these events two tasks have been addressed: 1) the retrieval of arguments on societal topics (e.g., climate change and electric cars) to provide assistance to users on searching for relevant pros and cons with which forming their own opinion; and 2) the retrieval of argumentative answers to individuals' personal decisions in everyday life expressed as comparative questions in the form "Is X better than Y with respect to Z?." As stated by the labs organizers, these tasks are of importance in community question answering websites such as Yahoo! Answers[1] and Quora[2], discussion forums such as Reddit[3], and debate portals such as DebateWise[4] and IDebate[5].

Independently of this trend, the automatic extraction of argumentative information from text collections has attracted researchers' attention in other fields. Specifically, in the late 2000s, *argument mining* was recognized as a research area with its own entity, emerging from the intersection of the computational linguistics (CL) and natural language processing (NLP) fields [4]. Hence, during the last decade, great advances have been done, ranging from the formulation of particular tasks and argumentative models to the creation of annotated corpora and the development of argument mining methods and tools [1]. In this context, research efforts have focused on domains such as legal documents, essays, news items and, more recently, social media content, where argument mining has been envisioned as a powerful tool for policy makers and researchers in social and political sciences [1]. Moreover, it has to be noted that researches have been mainly conducted on corpora in English.

Despite these advances, there is need for addressing other domains and dealing with corpora in other languages distinct to English. Hence, in this work, we explore the extraction and search of arguments in the e-participation domain –where citizens get involved in societal and governmental issues through digital tools. In particular, we aim to provide argument extraction and search functionalities for an e-participatory budgeting platform where residents post, comment and vote for proposals to address problems in their city, deciding how to allocate part of the municipal or public budget in a democratic deliberation and decision making process. The platform is Decide Madrid[6], whose user-generated textual contents are in Spanish.

For such purpose, we propose a general and flexible information retrieval framework which, in addition to text documents (i.e., citizen proposals) relevant to a given query, returns categorized and linked argumentative structures existing in or related to such documents within a collection.

[1] Yahoo! Answers, https://answers.yahoo.com

[2] Quora, https://es.quora.com

[3] Reddit, https://www.reddit.com

[4] DebateWise, https://debatewise.org

[5] IDebate, https://idebate.org

[6] Decide Madrid, https://decide.madrid.es

The framework is composed of a pipeline of modules targeting several tasks –text processing, argument-based annotation, argument mining, information retrieval, reranking and evaluation–, bridging the gap between work done by the information retrieval and argument mining research communities.

Additionally, the implementation of the framework for the above mentioned case study has brought novelties and valuable linguist resources: a taxonomy of argument relations, a lexicon of argumentative connectors (in English and Spanish), a corpus with argument-based annotations in Spanish, new argument extraction and retrieval methods, and an easy-to-use tool for argument-based annotation of text documents.

The remainder of the paper is structured as follows. Section 2 surveys related work on argument mining and argument search. Section 3 formalizes the addressed problem and introduces the considered case study. Next, Section 4 presents the proposed framework, detailing its implementation for the case study. Section 5 shows examples of outcomes and results obtained with the framework implementation. Finally, Section 6 ends with some conclusions and open research directions.

2. Related Work

In this section, we first provide a short overview of the argument mining area (subsection 2.1), introducing some of its main tasks, approaches and resources. Then, we survey recent work on the argument search problem in the information retrieval field (subsection 2.2).

2.1. Argument Mining

Argument mining is a research area aimed at developing computational methods to automatically extract arguments from natural language texts [5, 1]. Among others, it deals with three principal tasks, namely *detection of arguments* [6], *identification of argument components* [4], and *recognition of argument relations* [7].

The *detection of arguments* consists of splitting a text into argumentative and non-argumentative parts, each of them belonging to one or several (usually two) consecutive sentences [8]. The *identification of argument components* consists of classifying detected argumentative units into claims and associated premises and evidences [9]. Finally, the *recognition of argument relations* consists of identifying and classifying existing links between argument components, which express some form of support or attack.

Traditionally, these tasks have been conducted separately as classification problems using either heuristic techniques or feature-based machine learning models [8, 4]. Recently, by contrast, they have been started to be addressed jointly as NLP sequence labelling problems through embedding-based (deep) neural network models [10].

Despite their differences, both approaches share a challenging bottleneck: the scarcity of annotated argumentative corpora which may serve as training and testing data [1]. To deal with this limitation, efforts have been made on building datasets on certain domains, such as AIFdb [11] –a repository of databases e.g., AracuriaDB (with newspaper editorials, parliamentary records, court summaries, and panel discussions) and MM2012a (with transcripts from BBC Radio 4)–, IAC [12] –a corpus of political discussions from internet forums–, the ECHR corpus [13]

–a set of documents extracted from legal texts of the European Court of Human Rights–, and AAEC [14] –a corpus of persuasive essays–, among others. The majority of these datasets are composed of text collections in English.

Along with argument mining algorithms and datasets, progress has been made in the development of tools for creating and exploring structured argumentative data. Examples of these tools are argument graph editors (e.g., Agora,[7] Argunet,[8] DebateGraph[9] and Rationale Online[10]) and argument-based annotation platforms (e.g., Araucaria[11] and OVA[12]).

Finally, the argument mining research community has been actively involved in various events, such as COMMA,[13] the International Conference on Computational Models of Argument –annually organized since 2006–, ArgMining,[14], the International Workshop on Argument Mining –annually organized since 2014 in prestigious CL and NLP conferences like ACL, NAACL, COLING and EMNLP–, and specialized tutorials, like the ACL 2016 Tutorial on NLP Approaches to Computational Argumentation, the IJCAI 2016 Tutorial on Argument Mining, and the KI 2019-2020 Tutorial on Argumentation Technology.

As new contributions for the argument mining area, in our work, we have implemented and tested novel argument extraction methods, and have built a corpus on e-participation –a domain unexplored in the area– with argument annotations of citizen proposals and comments from an online forum-based platform in Spanish.

2.2. Argument Search

Argumentative information appears in a wide variety of documents on the web, such as blogs, discussion forums, news items, and reviews. Current search engines, however, do not support the effective retrieval of arguments. In addition to not being able to identify and extract arguments (and their components and relations) from textual content, they do not consider the relevance and quality of argumentative fragments according to aspects such as the controversy of discussed topics, the stakeholders involved in debates, the rhetorical, logical and dialectical characteristics of the arguments, the existence of opinion polarity biases, and the fairness and diversity of the retrieved argumentative information.

Motivated by this situation, in the last years, researchers have started to investigate new information retrieval approaches specialized in domains and applications where arguments represent the core of user information needs [15]. Hence, *argument search* (or *argument retrieval*) is being consolidated as a very relevant and promising research area. In this sense, while some developed argument search approaches make use of methods and resources from *argument mining* –where notable advances have been made since its origins in the late 2000s (as explained in subsection 2.1)–, the processes of indexing, ranking, summarization, visualization

[7] Agora collaborative argument visualizer, http://agora.gatech.edu

[8] Argunet argument map editor, https://sourceforge.net/projects/argunet

[9] DebateGraph argument network visualizer, https://debategraph.org

[10] Rationale argumentative map editor, https://www.rationaleonline.com

[11] Araucaria argument annotator, http://staff.computing.dundee.ac.uk/creed/araucaria

[12] OVA argument analyzer, http://ova.arg-tech.org

[13] COMMA, http://comma-conf.org

[14] ArgMining, https://aclanthology.org/venues/argmining

and evaluation of arguments in information retrieval tasks are underexplored, challenging problems.

In fact, some argument search tasks have been preliminary addressed [2, 16, 17] –such as identifying argumentation goals in a discourse, gathering premises to confront a given claim within an argumentative collection, finding arguments related to a controversial topic, or retrieving argumentative information to support decision making–, and there are others, as the one proposed in this paper, which can be formulated and considered for investigation.

Regardless of the targeted task, the argument search models and strategies can be classified into two major categories [17]: *text-based retrieval* (or *mining-before-retrieval*) and *argument ranking* (or *retrieval-before-mining*). Text-based retrieval approaches assume that argument mining is applied offline and that the extracted arguments are indexed for later online retrieval. Hence, these approaches make use of a standard search engine to retrieve documents related to a given query, and then extract and possibly rank the arguments of the top-scored retrieved documents. For instance, args.me [18] and ArgumenText [19] employ the BM25 model. Argument ranking approaches, by contrast, perform argument indexing and ranking operations, and exploit the outcomes of such operations through a specialized search engine. Examples of these operations are building argument graphs on which computing argument PageRank scores [20]), and clustering semantically similar claims and premises to identify groups of arguments related to the input query [16].

In addition to tasks and approaches, research work on argument search has also focused on the evaluation of arguments. Wachsmuth et al. [21] surveyed the argument quality dimensions considered in argumentation theory, and organized them within three categories: *rhetorical*, *logical* and *dialectical*. Arguments with high rhetorical quality are persuasive and appealing to the audience. Arguments with high logical quality contain acceptable premises and combine them in a convincing way to support the arguments' claims. Finally, arguments with high dialectical quality contribute to the discourse supporting decision making or conflict resolution. Potthast et al. [17] conducted a user study which showed that argument relevance and argument quality hardly correlate, and that rhetorical, logical and dialectical quality of arguments can be moderately distinguished by expert assessors, being dialectical quality the one with most in common with relevance. More recently and also aiming to complement topical relevance, Pathiyan et al. [22] explored the evaluation of fairness and diversity metrics to take into account possible biases of argument retrieval systems over positive or negative perspectives on controversial topics.

In a collaborative effort, the research community has organized and participated in the argument retrieval Touché labs[15], celebrated at CLEF 2020 and CLEF 2021, with 17 and 27 participating teams, respectively [2, 3], addressing the retrieval of arguments to support argumentative conversations and to answer comparative questions. In the first edition of Touché, submitted approaches shared common techniques, such as standard TF-IDF and BM25 retrieval models and query expansion techniques. The conducted evaluation showed that only a few of the approaches slightly improved upon relatively argumentation-agnostic baselines. Differently, in the second edition, submitted approaches improved upon argumentation-agnostic baselines for the two tasks. Most of them made use of the previous year Touché's data for parameter

[15] Touché labs, https://webis.de/events/touche-22/

optimization and model fine-tuning, showing an incipient interest in neural network-based solutions.

In this work, we first contribute to the area by proposing and formalizing a novel argument search task: retrieving both textual documents and associated arguments relevant for a given query. The goal of the task is to generate a summary of argued opinions with different polarities from input discussions and debates on certain (controversial) topic. To address this task, we propose a framework consisting of a pipeline of modules dealing with several information retrieval subtasks, such as text processing, argument annotation and extraction, and argument-based indexing, ranking and evaluation. Besides, as a proof of concept, we have developed and tested an implementation of the proposed framework for the above mentioned e-participation case study. Before presenting the framework, we next detail the problem formulation and case study.

3. Problem Formulation and Case Study

In this section, we formulate the argument search problem addressed by our framework (subsection 3.1) and the case study considered for the implementation of the framework (subsection 3.2).

3.1. Argument-based Document Retrieval

We propose a novel argument search task, which is the retrieval of text documents relevant to a given query, together with categorized and linked arguments related to such documents. The problem can be defined more formally as follows.

Let $\mathcal{D} = \{d_1, ..., d_N\}$ be a set of text documents of an input collection, and let $\mathcal{A}_n = \{a_{n,1}, ..., a_{n,L_n}\}$ be the set of arguments associated to document d_n. These arguments are assumed to be extracted by an argument mining method from the document itself or from other documents related to it. An argument a is defined as a tuple $a = (c, r, p)$ which is composed of a claim c and a premise p, linked to each other through a relation r of certain type of support or attack. Relations $r' \in \mathcal{A}_n \times \mathcal{A}_n$ between arguments could also be extracted for document d_n. In such a case, the set of arguments for a document d_n would form an argumentative tree or graph.

Given a keyword-based query q, the goal is to build a search model that generates a ranking function $score(d_n, q) \in \mathbb{R}, \forall d_n \in \mathcal{D}$, which would consider a similarity $sim(d_n, q) \in \mathbb{R}$ between a document d_n and the query q, and a relevance metric $rel(\mathcal{A}_n) \in \mathbb{R}$ for the arguments of d_n. Hence, the resultant model would retrieve a ranked list of documents, each of them accompanied by its arguments; that is, it would return a ranking of pairs $\{(d_n, \mathcal{A}_n)\}$.

The particular implementation of $sim(d_n, q)$ and $rel(\mathcal{A}_n)$, as well as their integration to compute $score(d_n, q)$, represent issues that call for research. While the similarity $sim(d_n, q)$ can be set with a classical information retrieval model (e.g., a vector space model), to the best of our knowledge, the relevance $rel(\mathcal{A}_n)$ represents an underexplored task in argument mining. In particular, we envision that it may consider argumentative aspects, such as the general polarity of given opinions and the degree of controversy within existing debates. Consequently, implementations of $score(d_n, q)$ are also open to investigation, and may range from function aggregation to document reranking approaches.

The proposed problem can be of interest for a variety of applications and domains. For instance, in the legal domain, a lawyer may need finding past court orders about a certain issue, and the argumentation derived from the associated trials. In a political context, journalists and politicians may need collecting transcripts of parliamentary debates related to a given topic, as well as the main arguments and counterarguments expressed within the MPs' interventions. In e-commerce applications, both customers and vendors may need obtaining informative reviews about certain products, as well as the underlying positive and negative opinions with the reasons for such opinions.

For all the above cases, the problem outcomes are summaries of opinions and arguments extracted from a text collection. In this sense, the classification, linking and visualization of arguments take on special importance, and consequently represent further research lines.

3.2. E-participatory Budgeting

As a proof of concept, we implemented and tested our argument-based search framework with the e-participation dataset published and analyzed in [23]. The dataset contains over 24.8K citizen proposals and 86.1K comments forming collective debates around the proposals, being rich in argumentative information. It is a crawled dump of Decide Madrid[16], the online platform of the annual participatory budgeting (PB) initiative of the City Council of Madrid, Spain, since 2014.

Participatory budgeting is considered as one of the major citizen participation approaches worldwide. It allows citizens to decide how to spend part of municipal or public budgets. In a PB process, people inform about issues and problems about a variety of subject areas (e.g., education, environment, housing, health, public safety, and transport), and propose, debate and vote for ideas and projects aimed to address such issues and problems. In general, after a period of time, those citizen proposals that receive more votes and are validated by government receive public funding and are implemented.

The Decide Madrid platform is built upon the CONSUL[17] framework, which has been made open source and, as of March 2022 has been used by at least 135 institutions of 35 countries supporting 90 million citizens around the world. Similarly to other electronic PB frameworks, such as the Stanford Participatory Budgeting[18] and the EU Open Budgets[19] tools, Decide Madrid follows the typical debate structure of online forums, which is composed of trees of hierarchical, nested comments.

More specifically, each citizen proposal has associated a tree. The root of the tree contains the proposal's title and description, whereas each of its nodes has a positive or negative (i.e., supporting or attacking) textual comment about the proposal or a parent comment in the tree. In the implementation of our framework, we aimed to automatically extract arguments from proposal descriptions and comments. Thus, for an input keyword-based query, the framework was targeted to return both citizen proposals (as documents) and linked, structured arguments associated to such proposals.

[16] Decide Madrid, https://decide.madrid.es

[17] CONSUL e-participation framework, http://consulproject.org

[18] Stanford Participatory Budgeting, https://pbstanford.org

[19] EU Open Budgets, http://openbudgets.eu/tools

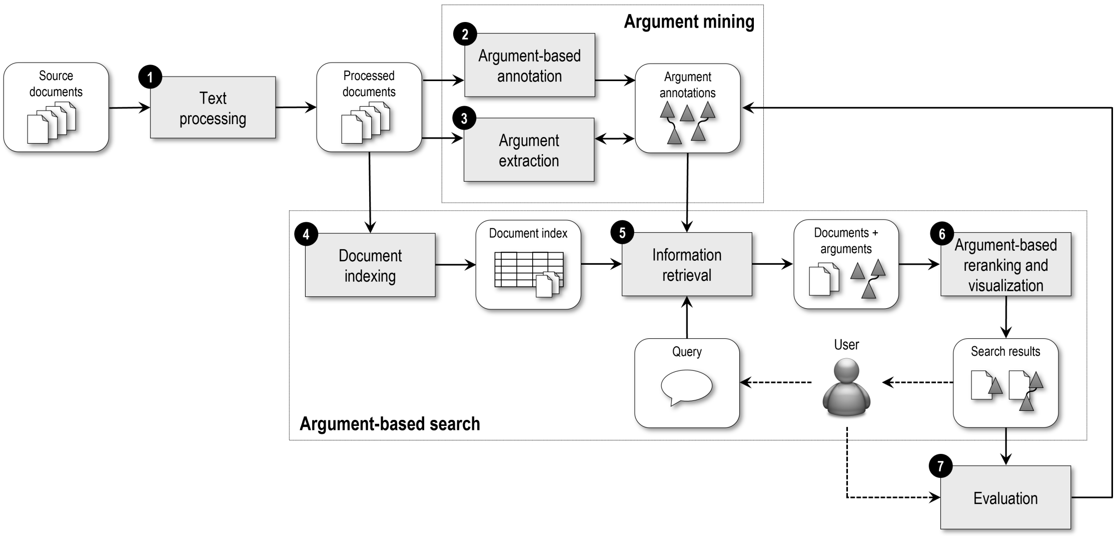


Figure 1: Modules and data flows of the proposed argument-based search framework.

From the original dataset, we limited our work to a set of 80 citizen proposals with 5,633 comments. These proposals were selected taking into account their topics and controversy, as measured in [23]. Hence, they uniformly covered 10 categories (i.e., animals, economy, education, equity & integration, mobility, natural environment, social rights, security & emergencies, sports, and urbanism), and were highly controversial.

4. Argument-based Search Framework

In this section, we present our generic and flexible argument-based search framework. We first give a general description of its modules and data flows (subsection 4.1), and then provide some details about its implementation for the considered e-participation case study (subsection 4.2).

4.1. Framework Description

The proposed argument-based search framework is composed of a pipeline of 7 modules organized in 2 logical blocks, namely *argument mining* and *argument-based search*. The framework allows the annotation, extraction, retrieval and validation of argumentative information from textual content. Figure 1 shows a comprehensive diagram of the framework. We next explain its modules' functionalities, inputs and outputs.

Text processing module. This module performs natural language processing on the source documents $\mathcal{D} = \{d_1, \cdots, d_N\}$ from which identifying and extracting arguments. It is in charge of splitting textual content into processable sentences and cleaning up the text, e.g., by removing hyperlinks, emoticons and contiguously duplicated punctuation marks, as well as correcting misspellings. Depending on the argument extraction method to be used afterwards, techniques for certain natural language processing tasks –such as part-of-speech (PoS) tagging, named entity recognition (NER), constituency and dependency parsing, etc.– could be also applied, in order to generate necessary linguistic features and metadata.

Argument-based annotation module. This module assists with the manual identification and annotation of arguments –each of them consisting of a tuple $a = (c, r, p)$ that relates a claim c and a premise p through a typed relation r– in processed sentences to generate training data that may be used by the used argument extraction method. This module is thus optional, depending on whether the argument extraction is based on supervised learning or employs a heuristic or self-supervised leaning approach.

Argument extraction module. This module wraps an argument extraction method, which is applied to the processed documents to automatically generate a set of well-formed arguments $\mathcal{A}_n = \{a_{n,1}, ..., a_{n,L_n}\}$ for each document $d_n \in \mathcal{D}$. Examples of argument extraction methods that can be implemented in this module are: i) rule-based and heuristic techniques that search for certain patterns within the syntactic trees of sentences, ii) traditional classifiers trained with previously extracted features/metadata and manually annotated argumentative phrases, and iii) (deep) neural network models based on embeddings, also built with a labeled training corpus, which address in an end-to-end fashion the principal argument mining tasks, namely *argument detection*, *argument constituent identification*, and *argument relation recognition*.

Document indexing module. This module creates an in-memory full-text index for the processed documents, oriented to optimize the information retrieval process. The index can be created from the title and textual content of the documents, as well as with some metadata, such as topics, categories, entities, etc., extracted from the documents. Argumentative information could be also considered, but we delegate its exploitation to the subsequent modules.

Information retrieval module. Given a keyword-based query, this module uses the full-text index –created by the previous module– to perform content-based filtering (e.g., using the well-known Boolean and Vector Space models) with which obtaining a (ranked) subset of documents that most closely match the input query. These documents are returned together with their respective arguments, $\{(d_n, \mathcal{A}_n)\}$. In addition to textual features, the underlying retrieval method could also exploit argument-based features to select and promote certain documents. If this is not the case, a subsequent module may apply a reranking strategy on the resultant document list according to argument-based aspects.

Argument-based reranking and visualization module. This module performs a reranking of the retrieved documents and arguments considering scores based on argument quantity and quality metrics, e.g., by combining the topic-based scores generated by the information retrieval module with certain controversy measurement. The module could also be in charge of displaying the documents accordingly.

Evaluation module. This module allows the user to review and validate the relevance and quality of the retrieved documents and arguments. It records user assessments in a structured form, storing them in a file or database. The outcomes of the evaluation could be further used to enhance the argument annotations, and consequently improve the argument extraction process.

4.2. Framework Implementation

To validate our framework, we implemented each of its modules, and tested them with the Decide Madrid dataset presented in subsection 3.2, composed of citizen proposals and comments in Spanish. We next explain the developed implementations[20].

Text processing. We conducted a number of data cleaning processes on the textual content of the Decide Madrid dataset. Some of the processes could be applicable to other user generated textual content in Spanish:

- Removal of hyperlinks.
- Capitalization of names of districts, neighborhoods and streets (from an open data repository) to facilitate their recognition as named entities.
- Transformation of slang abbreviations and acronyms, e.g., converting “ q ” to “ que ” (*que* is *what* in Spanish), and “ xq ” to “ porque ” (*porque* is *because* in Spanish).
- Transformation of grave accents into acute accents., e.g., converting ‘à’ to ‘á’.
- Addition of accents in interrogative and exclamative pronouns, e.g., converting “donde” to “dónde” (*dónde* is *where* in Spanish), and “como” to “cómo” (*cómo* is *how* in Spanish).
- Addition of accents in endings of certain verb tenses, e.g., converting “deberia” to “debería” (*debería* is *it should* in Spanish), and “podria” to “podría” (*podría* is *it could* in Spanish).
- Cleanup of sequences of contiguously repeated symbols and punctuation marks, e.g., replacing “!!!” by “!”.
- Addition of a blank space after each question/exclamation mark and certain punctuation mark: ‘.’, ‘,’, ‘:’ and ‘;’.

After these data cleaning processes, we made use of the Stanford CoreNLP library[21] to extract grammatical and syntactic metadata; specifically, by applying state-of-the-art natural language processing techniques for PoS tagging, NER and constituency parsing.

Argument-based annotation. We developed an easy-to-use Java tool[22] (Figure 2), which assists the user to identify and label arguments in input texts, and stores categorized claim-relation-premise tuples in a file or database as formal data structures (cf. Figure 3).

In particular, the tool allows the user to search and annotate argumentative information in the citizen proposals and comments. By highlighting part of a text, the user indicates an argument’s claim and premise. Subsequently, through a dialog window, the user selects the category (and sub-category) of the claim-premise relation, as well as its primary intention: *support* or *attack*. Moreover, the user can state the relevance and (rhetorical) quality of the annotated argument. More details on this latter aspect will be given in *Evaluation implementation* subsection.

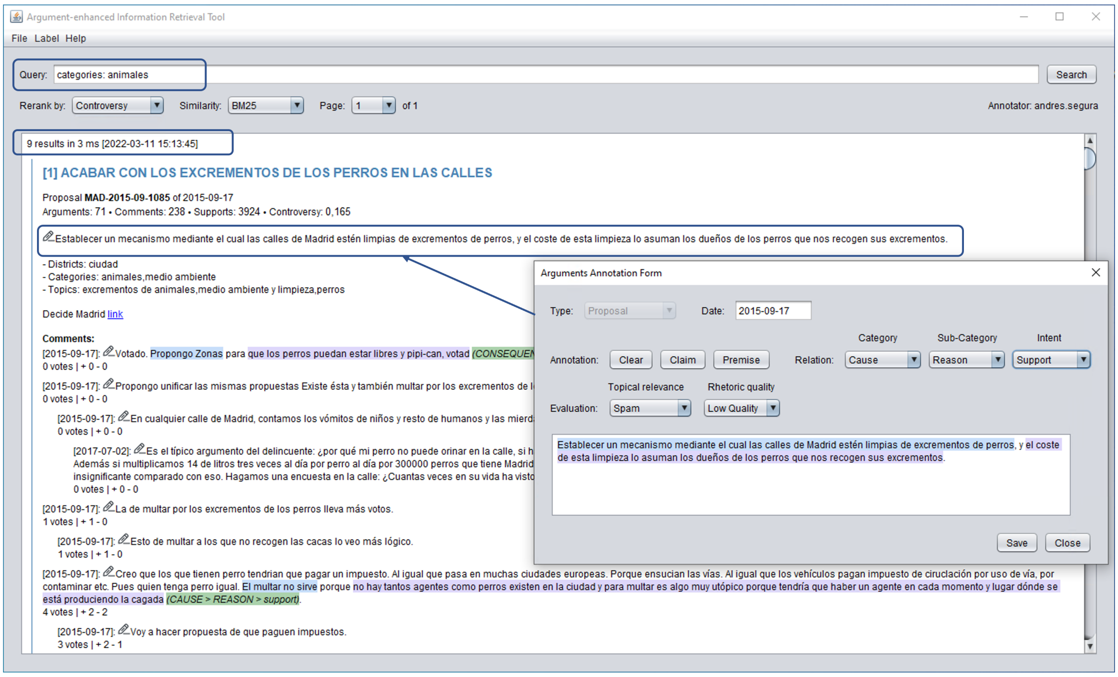
The above argument relation categories and sub-categories are the following:

- *Cause*: stating a premise that reflects the *reason* or *condition* for a claim. Example sentence: “There are monumental traffic congestion in Madrid, *because* public transport is not adapted to the current reality”.

[20] Source code available at https://github.com/argrecsys

[21] Stanford CoreNLP library, https://stanfordnlp.github.io/CoreNLP

[22] Argument-based annotation and search tool, https://github.com/argrecsys/annotator


Argument-enhanced Information Retrieval Tool
File Label Help
Search
Arguments Annotation Form
Clear
Claim
Premise
Save
Close

Query: categories: animales

Rerank by: Controversy Similarity: BM25 Page: 1 of 1 Annotator: andres.segura

9 results in 3 ms [2022-03-11 15:13:45]

[1] ACABAR CON LOS EXCREMENTOS DE LOS PERROS EN LAS CALLES

Proposal **MAD-2015-09-1085** of 2015-09-17
Arguments: 71 • Comments: 238 • Supports: 3924 • Controversy: 0,165

Establecer un mecanismo mediante el cual las calles de Madrid estén limpias de excrementos de perros, y el coste de esta limpieza lo asuman los dueños de los perros que nos recogen sus excrementos.

- Districts: ciudad
- Categories: animales,medio ambiente
- Topics: excrementos de animales,medio ambiente y limpieza,perros

Decide Madrid link

Comments:

[2015-09-17]: Votado. Propongo Zonas para que los perros puedan estar libres y pipi-can, votad *(CONSEQUEN*
0 votes | + 0 - 0

[2015-09-17]: Propongo unificar las mismas propuestas Existe ésta y también multar por los excrementos de l
0 votes | + 0 - 0

[2015-09-17]: En cualquier calle de Madrid, contamos los vómitos de niños y resto de humanos y las mierd
0 votes | + 0 - 0

[2017-07-02]: Es el típico argumento del delincuente: ¿por qué mi perro no puede orinar en la calle, si h
Además si multiplicamos 14 de litros tres veces al día por perro al día por 300000 perros que tiene Madrid
insignificante comparado con eso. Hagamos una encuesta en la calle: ¿Cuantas veces en su vida ha visto
0 votes | + 0 - 0

[2015-09-17]: La de multar por los excrementos de los perros lleva más votos.
1 votes | + 1 - 0

[2015-09-17]: Esto de multar a los que no recogen las cacas lo veo más lógico.
1 votes | + 1 - 0

[2015-09-17]: Creo que los que tienen perro tendrian que pagar un impuesto. Al igual que pasa en muchas ciudades europeas. Porque ensucian las vías. Al igual que los vehículos pagan impuesto de ciruclación por uso de vía, por contaminar etc. Pues quien tenga perro igual. El multar no sirve porque no hay tantos agentes como perros existen en la ciudad y para multar es algo muy utópico porque tendría que haber un agente en cada momento y lugar dónde se está produciendo la cagada *(CAUSE > REASON > support)*.
4 votes | + 2 - 2

[2015-09-17]: Voy a hacer propuesta de que paguen impuestos.
3 votes | + 2 - 1

Arguments Annotation Form

Type: Proposal Date: 2015-09-17

Annotation: Clear Claim Premise Relation: Category: Cause Sub-Category: Reason Intent: Support

Evaluation: Topical relevance: Spam Rhetoric quality: Low Quality

Establecer un mecanismo mediante el cual las calles de Madrid estén limpias de excrementos de perros, y el coste de esta limpieza lo asuman los dueños de los perros que nos recogen sus excrementos.

Figure 2: Argument-enhanced Information Retrieval tool.

- *Clarification*: introducing a *conclusion*, *exemplification*, *restatement* or *summary* of an argument. Example sentence: "We are on the way to a situation called the 'world upside down', *that is*, first the dogs and then the humans".
- *Consequence*: evidencing an *explanation*, *goal* or *result* of an argument. Example sentence: "Improve the horizontal and vertical signage in the city, *in order to* allow a traffic flow without incident".
- *Contrast*: attacking arguments by giving *alternatives*, doing *comparisons*, making *concessions*, and providing *oppositions*. Example sentence: "It seems to me a very accurate proposal, *although* selling it as a class struggle of rich and poor does not help".
- *Elaboration*: introducing an argument that provides details about another one, entailing *addition*, *precision* or *similarity* issues about the target argument. Example sentence: "I propose to place speed bumps in the avenues of Sanchinarro *as* a measure to limit the speed of cars and reduce the number of accidents".

Argument extraction. As a simple baseline, we implemented a heuristic method that automatically identify and extract arguments from processed documents without requiring training data.

The method looks for certain *argumentative patterns* in the syntactic tree of an input sentence through breadth first search (BFS). The patterns were defined by manual inspection of syntactic structures of sentences that have at least one *argument linker*,[23] i.e., a word or expression that is likely to connect claims and premises.

[23] Argument patterns and linkers, https://github.com/argrecsys/connectors

Once the method has found a syntactic pattern within a sentence, it proceeds to extract the corresponding claim (syntactic subtree on the left of the linker) and premise (syntactic subtree on the right of the linker). Below, we give three of considered syntactic patterns. The elements used in the patterns are: `[conj_LNK]` = conjunctive linker, `[grup.verb]` = verb group, `[neg]` = negation, `[S]` = clause, `[S_LNK]` = clause starting with a linker, and `[sn]` = noun phrase.

[S]-[conj_LNK]-[S]

`[sn]-[neg]-[grup.verb]-[S_LNK]`

`[grup.verb]-[sn]-[S_LNK]`

As an illustrative example, the sentence “We are almost forced to use public transport in the city, but pets are not allowed in EMT” satisfies the first of the above patterns, where *but* is the corresponding linker.

Since each linker is associated to a category (and sub-category) of our argument taxonomy, the method is able to automatically categorize the identified arguments. From the set of 80 citizen proposals with 5,633 comments, the method automatically extracted and annotated 1,744 arguments, of which 944 (54.1%) were of *Contrast* type, 525 (30.1%) of *Consequence* type, 211 (12.1%) of *Cause* type, 62 (3.6%) of *Elaboration* type, and 2 (0.1%) of *Clarification* type.

In addition to this heuristic method, we have also implemented a feature-based classifier (based on [4]) and an embedding-based deep neural network model (based on [10]), which require training data to be built. We do not explain them here since they are out of the scope of the paper.

Document indexing. Our Java tool integrates document indexing and retrieval implementations provided by the Apache Lucene library[24].

Specifically, we created a TF-IDF inverted index from the title and description of the citizen proposals, as well as some of their metadata, such as categories, topics, districts, neighborhoods and named entities. For instance, the indexed fields of the citizen proposal “Allowing pets in public transport”[25] are:

- *Title*: Allowing pets in public transport
- *Description*: We are almost forced to use public transport in the city, but pets are not allowed in EMT
- *Categories*: animals, environment, mobility
- *Topics*: pets, environment, public transport
- *Districts*: City
- *Entities*: EMT (which stands for “Empresa Municipal de Transportes de Madrid”, i.e., Madrid Regional Transport Company)

Information retrieval. Once the index has been created and all the arguments extracted from the citizen proposals and comments have been loaded into memory, our Java tool allows performing keyword-based queries (Figure 2). These queries can be stated as simple keywords

[24] Apache Lucene library, https://lucene.apache.org

[25] Proposal 5717 in Decide Madrid, https://decide.madrid.es/proposals/5717-permitir-mascotas-en-transporte-publico

(in this case, the searches are done over the proposals' titles) or by establishing "field: keyword" pairs, where the field can be proposal description (summary), categories, topics or entities, all of them corresponding to particular indexed fields. Examples of valid queries are:

- Q1: *"dogs"*, which retrieves 4 proposals along with 138 arguments.
- Q2: *"title: motorcycles OR title: bicycles"*, which retrieves 8 proposals along with 312 arguments.
- Q3: *"summary: Madrid"*, which retrieves 14 proposals along with 286 arguments.

The module retrieves the set of documents d_n (and associated arguments $\{\mathcal{A}_n\}$) satisfying the user query q, sorted by the $scores$ corresponding to the query-document relevance values. To compute these scores, the module can use the Boolean and Vector Space models, and the $Cosine$, $BM25$ and $DirichletLM$ similarities. The possible configurations can be chosen by the user in the tool (Figure 2), and are those that were employed in the Touché labs [2, 3].

Moreover, differently to previous work, as [19], our framework allows complementing the topic-based document retrieval with a reranking strategy exploiting argumentative information, as explained next.

Argument-based reranking and visualization. We implemented an argument-based reranking strategy that considers the argumentative controversy at document level. In particular, the strategy consists of a linear aggregation of the content-based $score$ returned by the information retrieval module, and a novel controversy metric [23]. Formally, being $\alpha \in [0,1]$, the final ranking score of a document is computed as:

$$arg\_score(d_n, q) = \alpha \cdot score(d_n, q) + (1 - \alpha) \cdot controversy(d_n)$$

In preliminary evaluations, through grid search, we set $\alpha = 0.35$ to foster argumentative and controversial content within the search results.

The controversy of a given citizen proposal (document) p is computed as a normalized aggregation of several scores which measure different aspects or notions of controversy.

$$controversy(p) = \frac{1}{3} \sum_{i=1}^{3} \frac{controversy_i(p)}{\arg\max_{p'} controversy_i(p')} \in [0,1]$$

Specifically, the implemented base controversy metrics are:

- *Discussion content-based controversy*: the length of the proposal's debate, measured as the sum of the length of its comments c.

$$controversy_1(p) = \sum_{c \in comments(p)} length(c)$$

- *Opinion polarization-based controversy*: a weighted ratio measuring the difference of positive and negative votes for the proposal's comments, being $pos(p)$ the sum of the number of positive votes and $neg(p)$ the sum of the number of negative votes given in the comments.

$$controversy_2(p) = 1 + \frac{min(pos(p), neg(p))^2}{max(pos(p), neg(p))}$$

- *Conversation structure-based controversy*: an adaptation of the H-index proposed by [24] for measuring discussion diversification, being H the Heaviside step function.

$$controversy_3(p) = \sum_{n=1}^{depth(p)} H(width(p, n) \geq n) + \frac{1}{1 + |comments(p)|}$$

Other metrics could be considered. In particular, we envision the possibility of measuring controversy in terms of the arguments obtained by the argument extraction methods. For such purpose, predicted relevance and quality, polarity and diversity of arguments are aspects that may be taken into account.

In the tool, apart from the reranking strategy, we integrated a visualization technique that displays retrieved proposals together with their comment trees following a traditional online forum representation, and highlighting the extracted arguments and their elements: claims in blue, premises in purple, and relations in green. Figure 2 shows the reranking results and visualization of argumentative information for the keyword-based query “categories: animales” (*animales* is *animals* in Spanish).

Evaluation. As the final module of the proposed framework, we implemented an argument evaluation component (i.e., a specialized dialog window) in the tool, which allows providing human judgements on arguments automatically extracted and retrieved by previous modules or manually created or edited by the user. In accordance to the state-of-the-art on argument evaluation (see subsection 2.2), we considered two types of judgments, namely *topical relevance* and *rhetoric quality*. Specifically, the labels available to assess the topical relevance $rel(\mathcal{A}_n)$ of an argument $\mathcal{A}_n$ were:

- *Very relevant*: an accurate and highly relevant argument with respect to the major claim of the discussion. Codified with the numeric value of 3.
- *Relevant.* An accurate, but moderately relevant argument. Codified as 2.
- *Not relevant.* A well-formed, but not relevant argument. Codified as 1.
- *Spam.* A false or poorly-formed argument. Codified as -1.

Regarding the rhetoric quality, which measures the effectiveness of an argument in persuading an audience, the available labels were:

- *High quality*: a strong persuasive argument. Codified as 2.
- *Sufficient quality*: an argument with sufficient strength to persuade someone. Codified as 1.
- *Low quality*: an argument with null or low persuasive capability. Codified as 0.

The tool stores the generated judgements on a database, for their later recovery, analysis and exploitation.

5. Outcomes and Results

In this section, we show examples of outcomes and results of our framework implementation. The extracted arguments are stored in JSON data objects for their later exploitation. For the citizen proposal “Allowing pets in public transport”, the argument extraction method automatically identifies an argument composed of the claim “We are almost forced to use public transport in the city” and the *contrast* premise “but pets are not allowed in EMT,” which attacks the proposal (major claim). Figure 3 shows in JSON format part of the argument structure associated to the given example. It contains: i) the identifier of the proposal, ii) the sentence where the argument was found, iii) the argument constituents, and iv) the linker (connector) and relation type, subtype and intent.

Figure 3: Part of the JSON object created for an argument that evidences a contrast premise on a proposal in favor of using Madrid public transport with pets.

```
{
  "proposalID": 5717,
  "majorClaim": "Allowing pets in public transport",
  "sentence": "We are almost forced to use public transport in the city
             but pets are not allowed in EMT",
  "claim": "We are almost forced to use public transport in the city",
  "premise": "pets are not allowed in EMT",
  "relationType": {
     "type": "CONTRAST", "subType": "OPPOSITION",
     "intent": "attack", "linker": "but" }
}
```

The extraction and retrieval of arguments from textual content also enables the possibility of finding argumentative threads associated to the documents, in particular, citizen proposals and their comments. Linked arguments can be interpreted as summaries of conversations aimed at debating certain ideas in favor or against a proposal or some of its aspects. To this end, the arguments found in a proposal and their respective comments can be represented and analyzed as a directed acyclic graph where argumentative threads can be found using the longest path algorithm. As an illustrative example, Figure 4 shows an argumentative thread obtained from the description and comments of a Decide Madrid proposal[26] related to the need of a “Massive tree planting in Madrid”.

As a preliminary offline evaluation, using the developed tool, we manually validated 20% of the arguments extracted by the simple syntactic pattern-based method. For the *topical relevance* metric, 8.6% of the arguments were labeled as *spam*, 36.9% as *not relevant*, 39.9% as *relevant*, and 14.6% as *very relevant*, whereas for the *rhetoric quality* metric, 42.3% of the arguments were of *low quality*, 40.6% of *sufficient quality*, and 17.1% of *high quality*. Although these results are modest, they can be considered acceptable as baseline values, taking into account they were obtained with a heuristic method that does not require training data and parameter tuning.

[26] Proposal 20389 in Decide Madrid,
https://decide.madrid.es/proposals/20389-arborizacion-masiva-en-madrid

Figure 4: Argumentative thread obtained from a citizen proposal. MC, C, P and R stand for major claim, claim, premise and relation, respectively.

```
> Root argument [depth level 0]:
  MC: Massive tree planting in Madrid.
  - Argument reply [depth level 1]:
    C: Planting trees native to the Madrid region.
    P: Improve air quality, maintain a natural lifestyle and improve urban
       aesthetics with living beings.
    R: {intent: SUPPORT, type: CONSEQUENCE, subType: GOAL}
    - Argument reply [depth level 2]:
      C: The first thing they should do is to stop cutting down healthy trees.
      P: They are doing in Manzanares neighborhood.
      R: {intent: SUPPORT, type: CAUSE, subType: REASON}
    - Argument reply [depth level 2]:
      C: More than 230 trees in 3 weeks with the excuse that they are very
         dangerous and will fall on us.
      P: When they started cutting down, only 4 of the 230 were hollow inside.
      R: {intent: ATTACK, type: CONTRAST, subType: OPPOSITION}
      - Argument reply [depth level 3]:
        C: If only the trees they cut down were replaced by younger ones.
        P: That is not the case.
        R: {intent: ATTACK, type: CONTRAST, subType: OPPOSITION}
```

6. Conclusions

In this paper, we have presented a general and flexible argument-based search framework, and have described its implementation and preliminary validation on a dataset with citizen proposals and debates generated in an online participatory platform.

The implementation includes several argument extraction methods, based on syntactic pattern matching, feature-based classification, and embedding-based deep neural network models. The two latter methods are supervised learning algorithms that require training data to be built. To assist on the manual generation of such labeled data, we have incorporated into the framework an easy-to-use tool for argument exploration, annotation and evaluation.

The document retrieval component of the framework was implemented upon traditional vector space-based models. The use of other models or complementary techniques, such as query expansion (as done in [2]), or the development of ad hoc argument-based document retrieval methods could be explored. In our implementation, we applied a reranking strategy that exploits certain controversy metrics. Alternative controversy notions, or other argumentative metrics (e.g., predicted relevance and quality, polarity and diversity of arguments) also represent open research issues. In this context, it will be interesting to investigate whether documents with high controversy (or even relevance) scores tend to have associated high-quality, valuable arguments.

Finally, the reported evaluation was preliminary and focused on a simple heuristic argument extraction method. In future experiments, more sophisticated argument mining approaches should be investigated. In this sense, additional metrics, e.g., the fairness and diversity of the extracted arguments [22] could be explored. Four such purpose, as stated before, the developed argument annotation tool would allow increasing the size of the used corpus and generating new ones.

Acknowledgments

This work was supported by the Spanish Ministry of Science and Innovation (PID2019-108965GB-I00).

References

[1] J. Lawrence, C. Reed, Argument mining: A survey, Computational Linguistics 45 (2020) 765–818.

[2] A. Bondarenko, M. Fröbe, M. Beloucif, et al., Overview of touché 2020: Argument retrieval, in: 11th CLEF, 2020, pp. 384–395.

[3] A. Bondarenko, L. Gienapp, M. Fröbe, et al., Overview of touché 2021: Argument retrieval, in: 12th CLEF, 2021, pp. 450–467.

[4] R. Mochales Palau, M.-F. Moens, Argumentation mining: The detection, classification and structure of arguments in text, in: 12th Conference on Artificial Intelligence and Law, 2009, pp. 98–107.

[5] R. Mochales Palau, M.-F. Moens, Argumentation mining, Artificial Intelligence and Law 19 (2011) 1–22.

[6] B. Hachey, C. Grover, Automatic legal text summarisation: Experiments with summary structuring, in: 10th Conference on AI and Law, 2005, pp. 75–84.

[7] M. P. García Villalba, P. Saint-Dizier, Some facets of argument mining for opinion analysis, in: 4th International Conference on Computational Models of Argument, 2012, pp. 23–34.

[8] M.-F. Moens, E. Boiy, R. Mochales Palau, C. Reed, Automatic detection of arguments in legal texts, in: 11th Conference on Artificial Intelligence and Law, 2007, pp. 225–230.

[9] R. Rinott, L. Dankin, C. Alzate, et al., Show me your evidence - An automatic method for context dependent evidence detection, in: 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 440–450.

[10] S. Eger, J. Daxenberger, I. Gurevych, Neural end-to-end learning for computational argumentation mining, in: 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 11–22.

[11] J. Lawrence, F. Bex, C. Reed, M. Snaith, AIFdb: Infrastructure for the argument web, in: 4th International Conference on Computational Models of Argument, 2012, pp. 515–516.

[12] M. A. Walker, J. E. F. Tree, P. Anand, et al., A corpus for research on deliberation and debate, in: 8th International Conference on Language Resources and Evaluation, 2012, pp. 812–817.

[13] R. Mochales Palau, A. Ieven, Creating an argumentation corpus: Do theories apply to real arguments? A case study on the legal argumentation of the echr, in: 12th Conference on Artificial Intelligence and Law, 2009, pp. 21–30.

[14] C. Stab, I. Gurevych, Annotating argument components and relations in persuasive essays, in: 25th International Conference on Computational Linguistics, 2014, pp. 1501–1510.

[15] R. Bergmann, R. Schenkel, L. Dumani, S. Ollinger, ReCAP - Information retrieval and case-based reasoning for robust deliberation and synthesis of arguments in the political discourse, in: 2018 Conference on Learning, Knowledge, Data, Analytics, 2018, pp. 49–60.

[16] L. Dumani, P. J. Neumann, R. Schenkel, A framework for argument retrieval, in: 42nd European Conf. on Information Retrieval, 2020, pp. 431–445.

[17] M. Potthast, L. Gienapp, F. Euchner, et al., Argument search: Assessing argument relevance, in: 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2019, pp. 1117–1120.

[18] H. Wachsmuth, M. Potthast, K. e. a. Al Khatib, Building an argument search engine for the web, in: 4th Argument Mining Workshop, 2017, pp. 49–59.

[19] C. Stab, J. Daxenberger, C. e. a. Stahlhut, ArgumenText: Searching for arguments in heterogeneous sources, in: 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, 2018, pp. 21–25.

[20] H. Wachsmuth, B. Stein, Y. Ajjour, "pagerank" for argument relevance, in: 15th Conference of the European Chapter of the Association for Computational Linguistics, 2017, pp. 1117–1127.

[21] H. Wachsmuth, N. Naderi, Y. e. a. Hou, Computational argumentation quality assessment in natural language, in: 15th Conference of the European Chapter of the Association for Computational Linguistics, 2017, pp. 176–187.

[22] S. Pathiyan Cherumanal, D. Spina, F. Scholer, W. B. Croft, Evaluating fairness in argument retrieval, in: 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 3363–3367.

[23] I. Cantador, M. E. Cortés-Cediel, M. Fernández, Exploiting open data to analyze discussion and controversy in online citizen participation, Information Processing & Management 57 (2020) 102301.

[24] N. Rethmeier, M. Hübner, L. Hennig, Learning comment controversy prediction in web discussions using incidentally supervised multi-task CNNs, in: 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, Brussels, Belgium, 2018, pp. 316–321.