Internet Argument Corpus 2.0: An SQL schema for Dialogic Social Media and the Corpora to go with it

Rob Abbott, Brian Ecker, Pranav Anand, Marilyn A. Walker
University of California, Santa Cruz
abby@soe.ucsc.edu, becker@ucsc.edu, panand@ucsc.edu, mawalker@ucsc.edu

Abstract

Large scale corpora have benefited many areas of research in natural language processing, but until recently, resources for dialogue have lagged behind. Now, with the emergence of large scale social media websites incorporating a threaded dialogue structure, content feedback, and self-annotation (such as stance labeling), there are valuable new corpora available to researchers. In previous work, we released the Internet Argument Corpus, one of the first larger scale resources available for opinion sharing dialogue. We now release the Internet Argument Corpus 2.0 (IAC 2.0) in the hope that others will find it as useful as we have. The IAC 2.0 provides more data than IAC 1.0 and organizes it using an extensible, repurposeable SQL schema. The database structure in conjunction with the associated code facilitates querying from and combining multiple dialogically structured data sources. The IAC 2.0 schema provides support for forum posts, quotations, markup (bold, italic, etc), and various annotations, including Stanford CoreNLP annotations. We demonstrate the generalizability of the schema by providing code to import the ConVote corpus.

Keywords: dialogue, argument mining, sentiment, stance, data integration, online forums, debate

1. Introduction

Large scale corpora have benefited many areas of research in natural language processing, but until recently, resources for dialogue have lagged behind. This is changing as more and more researchers work with social media sites structured in the form of dialogues, such as 4Forums, Create Debate and Reddit as well as sites such as Twitter that provide dialogic affordances such as synchronized community Tweet-Ups and the use of replies to other tweets (Abu-Jbara et al., 2012; Biran and Rambow, 2011; Somasundaran and Wiebe, 2009; Rosenthal and McKeown, 2015; Sridhar et al., 2015; Hassan et al., 2010; Cook et al., 2014; Cook et al., 2013; Bamman and Smith, 2015). In previous work, we released the Internet Argument Corpus, one of the first larger scale resources available for opinion sharing dialogue (Walker et al., 2012b).

Table 1: The size of each dataset included.

| Dataset          | Authors | Discussions | Posts | Tokens |
|------------------|---------|-------------|-------|--------|
| 4forums          | 3.5K    | 11K         | 414K  | 57M    |
| ConvinceMe       | 5.5K    | 5.4K        | 65K   | 6.5M   |
| CreateDebate     | 709     | 61          | 3K    | 275K   |

The IAC 2.0 corpus can support research on many different aspects of social language and dialogue structure. The language of dialogue, and particularly of conversations in online forums on social and political topics, is very different from newspaper articles or broadcast news. Subjective genres in traditional media tend to be both monologic and formal, while online debates are strongly dialogic, interpersonal, and colloquial, often containing emotional and colorful language, as exemplified by the excerpts in Fig. 1.

Fig. 1 illustrates, for example, the frequent use of discourse cues such as But, If and Because to mark discourse relations, e.g. comparison and contingency relations (Prasad et al., 2008). Dialog strategies in the corpus also often include rhetorical questions, which are intended to elicit responses by challenging another’s evidence or assumptions: And what is wrong with giving homosexuals the right to settle down with the person they love? (R3 in Fig. 1). Utterances may also be strongly emotional or highly rational, e.g. contrast What is it to you if a few limp-wrists get married in San Francisco? (R3 in Fig. 1) with It’s not a literal account unless you read it that way (R4 in Fig. 1). About 10% of the utterances in the corpus are sarcastic, e.g., Really? When I have a kid, I’ll be sure to just leave it in the woods, since it can apparently care for itself (R4 in Fig. 1) see also see Q2 and R2. Results are common: Here come the Christians, thinking they can know everything by guessing, and committing the genetic fallacy left and right (R5 in Fig. 1).

Much of the corpus is also labelled for STANCE, so that it is useful for studies on stance classification, i.e. whether the speaker is PRO or CON on an issue under discussion (Somasundaran and Wiebe, 2010; Somasundaran and Wiebe, 2009; Hassan et al., 2012; Murakami and Raymond, 2010; Hasan and Ng, 2013). Stance also interacts with agreement and disagreement classification (Yin et al., 2012; Rosenthal and McKeown, 2015), and argument mining, where it is useful to know the side of an issue that a particular argument supports (Misra et al., 2013; Hasan and Ng, 2014).

1IAC 2.0 is available at https://nlds.soe.ucsc.edu/lac2
In our own work to date, we have used the corpus for studies on distinguishing agreement and disagreement (Sridhar et al., 2014; Misra and Walker, 2015; Abbott et al., 2011), and to classify posts by stance-side (Walker et al., 2012a; Anand et al., 2011). We have put together a corpus of summaries of the dialogic threads of arguments on Gay Marriage (Misra et al., 2011) and to classify posts by stance-side (Walker et al., 2012a; Anand et al., 2011). We have put together a corpus of summaries of the dialogic threads of arguments on Gay Marriage (Misra et al., 2011), and done studies using these summaries as indicators of the importance of arguments and the facets of particular arguments that recur frequently across the corpus. We have also used this corpus to develop methods for extracting highly specific well-formed arguments on particular topics (Swanson et al., 2015).

Other to date that have used IAC 1.0 to recognize sarcasm and nastiness in dialogue [Lukin and Walker, 2013; Justo et al., 2014; Swanson et al., 2014], and to distinguish factual from emotional argumentation [Oraby et al., 2015]. Subcorpora of IAC 2.0 useful for working on these topics are also available for download [https://nlds.soe.ucsc.edu] Oraby et al.’s bootstrapped corpus of factual vs. feeling arguments [Oraby et al., 2015] and subsets of IAC labelled for disagreement and sarcasm have also been used by other researchers [Pavlick and Tetreault, 2016; Schlöder and Fernández, 2014; Joshi et al., 2015].

## 2. Internet Argument Corpus 2.0 Data

The IAC 2.0 data provides an expanded dataset consisting of dialogues from 4forums.com, CreateDebate.com, and Convinceme.Net. 4forums.com is an online forum for political debate and discussion. Its sub-forums cover a broad range of topics relevant to the US political landscape. Users may initiate discussion threads and respond to other posts. The ability to quote other posts in whole or in part is a common feature.
monly invoked mechanism which provides precise context.

IAC 1.0 contained only discussions from 4forums. The 4forums section of IAC 2.0 is based on a rescraping of 4forums, which resulted in 24,000 additional posts. However we now exclude discussions with only one author (primarily those with only one post) so the number of discussions dropped from 11,800 to 11,079. IAC 2.0 features improved unicode handling, expanded and improved topic annotations, simpler direct quotes (quotes now use their own text objects instead of referencing a segment within a post’s text object), supporting code in Python3 instead of Python2. Most importantly the dialogues are organized into an SQL schema, as presented in this paper, instead of JSON and CSV format.

The IAC 2.0 4forums dataset consists of 414,453 posts in 11,079 threads by 3452 authors and 56M tokens. We have a number of annotations for this site including topic (2894 discussions), author stance (2248 authors by topic), agreement, sarcasm, and hostility measures (9975 quote-response pairs).

ConvinceMe. In addition to the 4forums dataset, IAC 2.0 includes dialogues from ConvinceMe.net, a highly structured debate site. This is an expanded version of the data used in (Anand et al., 2011) and (Walker et al., 2012d). The dataset consists of 65,368 posts in 5413 debates by 5783 authors. Users may initiate a debate by specifying the topic and sides. Other users are then forced to self-label stance when commenting by posting on the side they support or by using a rebuttal mechanism, which forces their post to the opposite side.

By choosing which side to post on authors self-label for stance. We annotated discussions for topic and mapped the discussion stance to a broader topic stance. Noting the lack of a human topline for stance classification in previous work (Somasundaran and Wiebe, 2010) (Thomas et al., 2006), we collected human topline stance annotations for this corpus (Anand et al., 2011). The annotation task presented the annotators with the topic, sides, and a sample post from each side, and then asked them to decide which side of the debate a post belonged to. Context, such as a parent post, was not provided, because this most nearly approximated the conditions under which automatic algorithms for stance classification operated at that time. These annotations are included in the release.

CreateDebate. We also introduce a gun control specific subset of CreateDebate.com, a debate site that, like ConvinceMe, exhibits a highly structured two-sided format. The subset consists of 2958 posts, 16,671 sentences, and 275,472 tokens. Similarly to ConvinceMe, the user starting the debate defines the topic, an introduction post, and the sides/stances to the debate. Top level posts are placed in the left or right column based on their stance. Responding posts must label their stance from the available sides when posting and must respond with a support, clarify, or dispute tag. Unlike ConvinceMe, responses on CreateDebate appear inline under their parents, creating a more natural discourse. It is also possible for a user to dispute the post of another user even if they self-label the same stance, which creates the opportunity to analyze how debaters supporting the same stance may disagree on certain sub-issues (Sridhar et al., 2014). Like ConvinceMe, CreateDebate allows users to vote on other posts, which the dataset also includes. These votes have been used to analyze persuasion effectiveness (Jaeck et al., 2015).

There are other releases of subsets of CreateDebate (Hasan and Ng, 2013; Rosenthal and McKeown, 2015). We believe that the IAC 2.0 schema applied to CreateDebate provides a more complete representation of CreateDebate’s extensive affordances and post-response system. It is possible to import other CreateDebate subsets into the IAC 2.0 schema.

Other Datasets. We have used this schema successfully with the ConVote corpus as well as with data from Twitter and Reddit. We provide code to import ConVote with all its...
3. Database Description

The SQL database provides a single consistent storage location for all data relevant to a given dataset including text, metadata, parses, annotations and partial computations. Extensive foreign keys help to explain relationships within a dataset and ensure referential integrity. Using SQL means that basic tasks can be accomplished with a simple query instead of running code and iterating over the dataset. By using the same schema for multiple datasets it is not only easier for humans to understand those datasets but also enables a single codebase with minimal dataset specific code. Using SQL also gives the option of non-local storage & access (client-server).

Because data that would be otherwise stored together in an XML file is scattered across several tables, some straightforward queries may involve several joins. Thus, for instance, to find sentences in a topic area, one must join a clutch of tables together (posts, discussions, topics, texts, and sentences). However, this is balanced by the fact that we do not have to loop over the entire dataset to find objects of interest.

In general the schema attempts to minimize redundancy, use foreign keys where possible, use consistent column names (thereby enabling NATURAL JOINs and the USING keyword), and prefer fixed width tables. Composite primary keys are used throughout and usually contain one or more foreign keys. For more complex tasks we provide a Python3 interface using an SQLAlchemy based ORM.

**Basic Structures.** The core portion of the schema consists of tables for discussions, posts, authors, quotes, and texts. Fig. 4 illustrates a minimal set of tables and columns which affordances (Thomas et al., 2006).
new datasets should provide to be importable. Additional dataset specific columns or even tables may be included (e.g. author.isi_language, post.votes, etc.). We also provide an SQL view (virtual table) joining the posts table with the others in order to reduce boilerplate and make the schema more user friendly. See Fig. [5]

Rather than putting post text in the posts table we store it in a dedicated table, because there are text strings outside of posts (discussion introductory blurbs) as well as inside posts (quotes) that we wish to store, and a unified location simplifies that task. This also allows markup and annotation tables (including the CoreNLP tables) to reference the text regardless of its source.

Quotes are an important affordance of the IAC. While quotes typically come verbatim from previous posts, they are ultimately a form of markup, and users often alter quoted material or quote from posts elsewhere in a discussion or on the site as well as external sources (e.g., Wikipedia). We store quote information as standoff annotation in the quotes table. If a quote’s original source post can be identified, it is referenced by identifiers and a text offset. We also mark differences between the quote and its source.

Parses. The schema provides support for Stanford CoreNLP [Manning et al., 2014] annotations including tokenization, part of speech tags, parse trees, dependencies, named entities, coreference, and sentence level sentiment. Scripts are provided for calling CoreNLP to generate xml output and storing the parses in the database. We store the constituency parses in a nested set data structure which supports queries over parse structures. See Fig. [6]

Annotations. There are also a large number of annotations for the corpus with additional annotations being added all the time. See Fig. [7]

4. Examples

Simple Query. This query finds sentences in a particular topic area. It can be simplified further using a view that combines the tables relevant to posts.

query = "SELECT HOUR(timestamp) AS AS_hour, COUNT(*) AS AS_count FROM posts GROUP_BY_HOUR( timestamp );"

data = dbGetQuery(conn = con, statement = query)
plot(data, type='l', ylim=c(0, max(data$count)), main='Posts by Hour')

Figure 8: This plot illustrates accessing the database using R (see example code). It shows activity on 4forums as a function of time.

Iterate over posts using Python. This example illustrates using the provided Python code to iterate over post objects printing their text and dependancies. We use SQLAlchemy’s ORM to map the database to Python objects.

from iac_objects import *
dataset = load_dataset('fourforums')
for discussion in dataset.get_discussions():
    for post in discussion.get_posts():
        print(post.full_id(), post.text)
        post.load_parse_data()
    for dep in post.text_obj.dependencies:
        print(dep)

Noun Phrases Parses Query. This query finds all noun phrases from within posts.

SELECT SUBSTR(text, MIN(start)+1, MAX(end)-MIN(start)) AS np_string FROM corenlp_pares,
NATURAL JOIN parseTags
NATURAL JOIN texts
NATURAL JOIN posts
JOIN tokens USING(text_id)
WHERE parse_tag='NP'
    AND tokens.node_index >= corenlp_pares.node_index
    AND tokens.node_index <= descendant_right_index
GROUP BY

library (RMySQL)
sql_access = #loads access info ...
con = dbConnect(MySQL(),
    user=sq_access$username, password=sq_access$password, host=sq_access$host, dbname=sq_access$database)
Dependency Parses Query. This query finds all dependencies which are within noun phrases. The selection is also limited to posts thereby excluding quotes or other text sources. It makes use of the dependencies_view which reduces the boilerplate code pulling in the governor and dependent tokens as well as various reference tables.

```sql
SELECT relation, governor_word, dependent_word FROM dependencies_view NATURAL JOIN posts  -- Only posts, not quotes NATURAL JOIN corenlp_parses NATURAL JOIN parseTags WHERE parse_tag='NP' AND dependent_node_index >= node_index AND dependent_node_index <= descendant_right_index AND governor_node_index >= node_index AND governor_node_index <= descendant_right_index;
```

Annotation Query. This query pulls out Quote-Response pairs with annotations as used in Fig. 1. The list it returns is sorted so that particularly sarcastic posts appear first.

```sql
SELECT topic, disagree_agree, attacking_respectful, emotion_fact, nasty_nice, sarcasm_yes, quote_text text AS quote,  -- the response may be only a portion of the post's text so we use substr() SUBSTR(texts.text, text_offset + 1, IF(response_text_end IS NOT NULL, response_text_end - text_offset, LENGTH(texts.text))) AS response FROM mturk_2010_qr_entries NATURAL JOIN mturk_2010_qr_task1_average_responses NATURAL JOIN posts NATURAL JOIN texts  -- Can't NATURAL JOIN quotes since its text_id is not the same as posts JOIN quotes USING(discussion_id, post_id, quote_index) JOIN texts AS quote_text ON quotes.text_id=quote_text.text_id ORDER BY sarcasm_yes DESC;
```
5. Acknowledgements

This work was funded by NSF Grant IIS-1302668-002 under the Robust Intelligence Program. The collection and annotation of the IAC corpus was supported by an award from NPS-BAA-03 to UCSC and an IARPA Grant under the Social Constructs in Language Program to UCSC by subcontract from the University of Maryland.

6. Bibliographical References

Abbott, R., Walker, M., Jean E. Fox Tree, Anand, P., Bowman, R., and King, J. (2011). How can you say such things?!?: Recognizing Disagreement in Informal Political Argument. In Proc. of the ACL Workshop on Language and Social Media.

Abu-Jbara, A., Diab, M., Dasigi, P., and Radev, D. (2012). Subgroup detection in ideological discussions. In Proc. of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1, ACL ’12, pp. 399–409.

Anand, P., Walker, M., Abbott, R., Jean E. Fox Tree, Bowman, R., and Minor, M. (2011). Cats Rule and Dogs Drool: Classifying Stance in Online Debate. In Proc. of the ACL Workshop on Sentiment and Subjectivity.

Bamman, D. and Smith, N. A. (2015). Contextualized sarcasm detection on twitter. In Ninth Int. AAAI Conf. on Web and Social Media.

Biran, O. and Rambow, O. (2011). Identifying justifications in written dialogs. In 2011 Fifth IEEE Int. Conf. on Semantic Computing (ICSC), pp. 162–168. IEEE.

Boltuzic, F. and Šnajder, J. (2014). Back up your stance: Recognizing arguments in online discussions. In Proc. of the First Workshop on Argumentation Mining, pp. 49–58.

Conrad, A., Wiebe, J., et al. (2012). Recognizing arguing subjectivity and argument tags. In Proc. of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics, pp. 80–88. Association for Computational Linguistics.

Cook, J., Kenthapadi, K., and Mishra, N. (2013). Group chats on twitter. In Proc. of the 22nd international conference on World Wide Web, pp. 225–236. Int. World Wide Web Conf. s Steering Committee.

Cook, J., Das, A., Kenthapadi, K., and Mishra, N. (2014). Ranking twitter discussion groups. In Proc. of the second edition of the ACM conference on Online social networks, pp. 177–190. ACM.

Habernal, I. and Gurevych, I. (2015). Exploiting debate portals for semi-supervised argumentation mining in user-generated web discourse. In Proc. of the 2015 Conf. on Empirical Methods in Natural Language Processing, pp. 2127–2137.

Habernal, I. and Gurevych, I. (2016). Argumentation mining in user-generated web discourse. CoRR,
