Multi-modal Multi-threaded Online Forums

Adam Wyner**

Department of Computer Science
University College London
Gower Street, London WC1E 6BT, United Kingdom
adam@wyner.info
wyner.info/LanguageLogicLawSoftware

Abstract. This paper motivates and outlines multi-modal multi-threaded online forums. The basic idea is that rather than a single "reply" mode as in current multi-threaded online forums, where the user can only indicate that a current input statement is a reply to an existing statement, the user can select from a menu among a range of ways the input statement relates to a particular existing statement. For example, in the menu might be "contradicts", "is premise of", "is conclusion of", or "is an exception to" relations. From such choices, one can construct an "argument dialogue". The input statements are parsed and semantically interpreted by the Attempto Controlled Language, ensuring grammatically well-formed statements with first order logic representations. The database of propositions and their relations can be used by further systems.

Key words: online forums, argumentation

1 Introduction

Multi-threaded online forums (MTOF) have proven to be a popular way for people to exchange information, whether to comment on a blog or to exchange information on a discussion board. They are easy to install, maintain, and use. In multi-threaded forums, users can recusively respond to particular topics, building a tree of topics and comments. Standardly, users only need to indicate that they wish to respond to a topic in order to open a text box for text entry, which is then saved.

While MTOFs clearly have advantages in terms of gathering and storing information, they are not, in and of themselves highly *informative*. Rather, they require a great deal of manual analysis to allow extraction of information, inference, or argument mapping. For example, consider the BBC's *Have Your Say* section, in which a problem or topic of discussion is presented on line for members of the public to respond to. People then write their comments in. However, the *semantic relationships* between the statements is not, in many cases, made explicit, but must be inferred. While one comment

may refer to an immediately preceding comment or give a number or other index, often this is not the case. Consequently, it is difficult to reconstruct the overall argument, the relationships among statements, and the conclusions that might follow from the give set of statements. Indeed, the longer the list of comments, the more problematic the task. In addition, it is difficult to prevent or filter out redundancy.

To make a more useful multi-threaded dsicussion list, we suggest a *multi-modal* multi-threaded discussion list (MMMTOF). In current MTOF, there is a single "reply" mode, where the user only indicates that a current input statement is a reply to an existing statement. In a MMMTOF, the user can select from a menu among a range of ways the input statement relates to a particular existing statement. For example, in the menu might be "contradicts", "is premise of", "is conclusion of", or "is an exception to" relations. From such choices, one can construct an "argument dialogue".

In addition to modes, we propose that the input statements are parsed and semantically interpreted by the Attempto Controlled English (ACE), ensuring grammatically well-formed statements with first order logic representations as well as some of their discourse referents. Given well-formed propositions, we have a database of propositions and their relations which can be used by further systems such as an *argumentation framework* such as [1] or argument graphing systems such as *Araucaria* [2] and *Carneades* [3].

While ACE as a controlled language is more limited in certain respects, it is nonetheless highly expressive and flexible, allowing users to enter natural and grammatical English sentences. While using an MMMTOF may not be for just any user with any sort of English, in the context of valuable policy discussions, the advantages strongly outweigh the disadvantages.

In the following, we give an example, background, motivation, and an initial specification. Our target MMMTOF is to be expressed using MySQL and PHP, based on an existing, open source, off-the-shelf MTOF [4].

2 Example

The example is derived from a public discussion list concerning recycling. We are considering a restricted domain and normalising the language; it is not our intention to address issues related to unrestricted domains or the full range of linguistic forms. We follow approach to argument graphing as in [5], where ordinary legal disputes are formalised. Each statement is represented as a node, claims and premises are represented with continuous arrows between nodes, while contradictions or conflicts between statements are represented with dashed arrows. The question that is under discussion is: *How can a government reduce the amount of garbage*? Each statement is made separately by an individual on the discussion list, and the order of introducing a statement in the discussion list may be different from the order below. We have graphed the relationships between the statements in Figure 1.

1. Every householder should pay tax for the garbage which the householder throws away.
2. No householder should pay tax for the garbage which the householder throws away.

3. Paying tax for garbage increases recycling.
4. Recycling more is good.
5. Paying tax for garbage is unfair.
6. Every householder should be charged equally.
7. Every householder who takes benefits does not recycle.
8. Every householder who does not take benefits pays for every householder who does take benefits.
9. Professor Resicke says that recycling reduces the need for new garbage dumps.
10. A reduction of the need for new garbage dumps is good.
11. Professor Resicke is not objective.
12. Professor Resicke owns a recycling company.
13. A person who owns a recycling company earns money from recycling.
14. Supermarkets create garbage.
15. Supermarkets should pay tax.
16. Supermarkets pass the taxes for the garbage to the consumer.

For example, an individual makes statement [1], another makes [4] as a reason or premise for [1], yet another makes [3] as an additional reason for [3], which can be understood to lend greater strength to the claim that [1] should hold. [9] supports the claim in [4]. However, this is undercut by the claim that the Professor is not objective, so the implication one might draw from his statement does not hold. In [2], we have a counter-proposal with a range of supporting reasons; the counter-proposal can be understood as a rebuttal to the previous argument in favour of taxing garbage. [16] attacks [15], which is one of the premises of the argument in favour of [2], so constitutes a premise defeat.

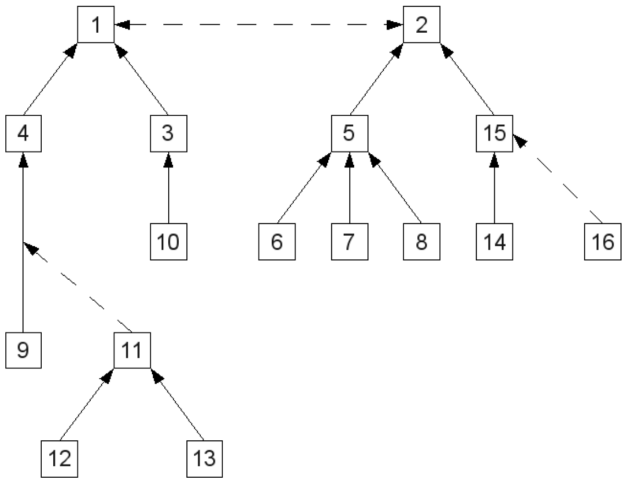

Fig. 1.

This "discussion" is not to be taken as a full, complete representation of all possible statements and counter-statements. Moreover, in some cases, there is an intuition that one statement attacks another statement – [16] attacks [15] – but much is left implicit. Argumentative discussion proceeds by such partial steps with missing premises.

From such a representation of the relationships among the propositions, we would like to be able to calculate consistent sets of propositions as in *argumentation frameworks* [1] as well as the "winning" proposition. We need, then, the means to control and structure the natural language input so as to support the logical translation.

3 Background

3.1 Access to BBC "Have Your Say" Discussions

The BBC has a range of current Have Your Say and for the most recent. For example, see Is a degree still worthwhile?. Each discussion has an RSS feed, where *numItems*= in the http address varies the number of messages one has access to, for example, returns 100 items.

In addition to such public policy discussions in comment form, governments regularly have *policy consultations with the public* which can be found.

3.2 Search and Information Extraction

Despite the lack of structure, there have been efforts to apply information extraction techniques to the contents of blogs and their comments in order to determine relationships among postings. [6] is an overview of blog search engines and their search capabilities, which are limited to familiar search capabilities such as date, phrase control (e.g. exact match, inclusion or exclusion of words), and specified headers (information in the blog title, URL, or other indices). Some (e.g. Google Blog Search) have *meme trackers* which allow search for and organization of blogs according to concepts or terms that they share. [7] and [8] show how subjects of online debates can be successfully tracked in short documents according to co-occurring terms. However, performance falls in relation to the length of each document. [9] discuss sentiment analysis in online forums.

In other literature [10], there has been some limited success in identifying argument structure (premises, rule, claim) from texts so long as overt indicators of argument structure are present (e.g. indicators such as *supposing* and *therefore*). Performance falls where argument structure must be inferred. So far as we know, there are no studies of forums or blogs which identify the point-counterpoint structure that one might find in a debate.

Current forums contain a great deal of information, and yet they are relatively uninformative in the sense that it is difficult to glean much useful information from them without detailed reading, review, and summarising. Moreover, the forums require the reader to interpret the relationships among the posts and infer the meanings using their linguistic capacities. While people are capable of this, we are a ways away from having machine support for this.

4 Motivation

Given that forums are currently relatively uninformative, we propose a system of incremental development of forums: initial forums will have no NLP backend and will have a simple set of context-sensitive alternatives from among which the user selects, indicating what the user intends the relationship between the posts to be. These alternatives are specified by a protocol; most any protocol can be provided in a general system. Given that we are not applying an NLP backend, a user can in principle still enter arbitrary text just as with similarly unconstrained forums. For example, suppose someone enters in *Bill is happy*. The next user simply wants to contradict this statement. Intuitively, the user would likely enter *Bill is not happy*. Yet, given that any string can be entered, the user could input *Colourless green ideas sleep furiously*. Moreover, even were the user to enter *Bill is not happy*, we are not making use of NLP to parse and semantically interpret the sentences, so there is no overt machine-readable indication that the statements are in contradiction. To overcome this limitation, we suppose that the user overtly marks the intended relationship between the statements to be contradiction: rather than simply *reply*, the user tags the statement with *contradicts*, then enters the contradiction itself, namely *Bill is not happy*. From the user's point of view, there is some redundancy, but form the point of view of machine processing, there is not: *Bill is happy* is some proposition say P, *Bill is not happy* is some other proposition say R, and we have explicitly introduced the *contradiction relation* between them: *contradiction(P, R)*. We can use this information for further semantic processing.

By the same token, we can introduce other semantically valuable relational markings between statements entered in the forum. Keeping to a rather coarse-grained approach, we want relational indicators to support argumentation. Therefore, we use indicators to signal premises, exceptions, rules, and claims. Given this, the statements and their relationships could be processed further.

Moreover, we envisage that the tool is used as an input to interface with systems that provide a syntactic parse and semantic interpretation such as *ACE* or *C&C with Boxer*. In addition, the tool would be used to provide input for an expert system that generates questions and queries users for the truth value of statements and can, then, be used to draw implications. There are three main advantages to these system. The first is that users can modify arguments dynamically over the internet. The second is that the parts of the argument are overtly represented, which can subsequently be used in an argumentation system. The third is that the system distinguishes the natural language input along with semantic relationships from how those relationships are treated by a logical system; in particular, we rely on users to indicate the relationship of contradiction rather than having that appear in the logic. In a system based on DRT, contradiction is a logical property and gives rise to inconsistency. As it is essential to represent contradiction in argumentation, but not simply be left with inconsistency, we choose to linguistically represent contradiction, leaving its semantic consequence for another component. Taken together, the tool allows an overt representation of argument in natural language which is then suitable for input into a formal argumentation system and for the calculation of consistent sets of arguments.

We have used *ACE* to provide first order logic translations of all the statements in 1-16.

In terms of our target audience, we have a highly constrained initial system targetted at a limitied audience in a high value discussion context such as law, politics, medicine, science, or the military, where the course and content of debate is crucial, a record of debate is useful, the content is already significantly formal, and the users are already disposed to some formal discourse. We suppose that users must login to the system and must undergo some initial training in the use of the system, though the learning curve is intended to be very low. It is unlikely that the forum would be suitable to the general public.

5 Implementation

5.1 Constraints

With these uses in mind, users must abide by some use conditions. This is a proposed specification and subject to comment and modification.

- The input abides by the ACE input conventions (described below). However, as the input sentences are not parsed by ACE, the user provides input meeting the conventions as best they can.
- Each input statement is one sentence without logical connectives other than negation. Rules are built interactively with the tool.
- The user builds rules (premises, rule, claim) which intuitively make sense as rules.
- Contradictions should be expressed as negations of one another, e.g. Bill is happy versus Bill is not happy. There may be syntactic or morphological variants of contractictions, but they must abide by the ACE conventions.
- Premises of a rule are added sequentially and are interpreted as conjunction, e.g. given premises Bill is rich, Bill is healthy, and the claim Bill is happy, we understand this as if Bill is rich and Bill is healthy, then Bill is happy. Premises must be atomic statements (no conjunctions, disjunctions), but this allows for negations: Bill is rich, Bill is not married, Bill is healthy, therefore Bill is happy. For the purposes of this version, we have no rules with disjuncts.
- Since we are not yet using ACE, the vocabulary is not restricted.
- If a statement is the claim of two or more rules, we take this as a disjunctive implication: If Bill is rich, then Bill is happy; If Bill is healthy, then Bill is happy; If Bill is rich or Bill is healthy, then Bill is happy.

Could have a simpler preliminary system where we just indicate agree or disagree with a previous post and give our reasons why.

Could have a more complicated system which implements the Wigmore argument graphs (warrants etc) for legal argumentation.

Other protocols to determine other relationships among statements.

5.2 Databases

We have a database for *statements*. The *statement_form* is the input. At this point, we do not consider the parse or the semantics, which are provided by ACE or similar.

- statement_id
- statement_form
- statement_parse
- statement_semantics (how to relate sentences, e.g. pronouns etc. First pass, not to worry.)
- stated_by
- stated_when

Users have the following DB:

- user_id
- statement_id

We have rules of two sorts:

- rule_id
- rule_sort (e.g. defeasible or strict)

Arguments are constructed from rules, claims, premises, and exceptions:

- argument_id
- rule_id
- claim (single statement)
- premises (one or more statements)
- exceptions (one or more statements)

Contradictions are pairs of statements. No statement can contradict itself.

- contradiction_id
- statement_id
- statement_id

5.3 Protocol

The tool provides a protocol for explicitly indicating the relationships among statements.

For any statement that is the current statement, the user can contradict that statement. Each statement is entered into the database of contradictions, which is understood to be reciprocal; if statements A, B appear, then A contradicts B and B contradicts A.

- Current statement: Bill is happy.
- Add: Bill is not happy.

For any statement that is the current statement, the user can make that statement the claim of a new rule. The user is asked to select a rule sort (defeasible or strict) and to input a premise. Additional premises are interpreted conjunctively – they all must be true for the claim to hold.

- Current statement: Bill is happy.
- Add: Bill is happy presumably if Bill is rich.
- Alternatively add: Bill is happy strictly if Bill is healthy.

  For any statement that is the current statement and the statement is already a claim of a rule, the user can add a premise to that rule.

- Current statement which is the claim of a rule: Bill is happy (presumably if Bill is rich).
- Add: Bill is unmarried (as premise of the rule).

  For any statement that is the current statement, the user can make that statement the premise of a rule. The user is asked to select a rule sort and the claim.

- Current statement: Bill is happy.
- Add: Bill makes other people happy (presumably if Bill is happy).

  For any statement that is the current statement and the statement is the claim of a rule (so therefore has at least one premise), the user can add an exception. Additional exceptions are interpreted conjunctively – they all must be false (and the premises all true) for the claim to be true.

- Current statement which is the claim of a rule: Bill is happy (presumably if Bill is rich).
- Add: All Bill's money is in volatile assets (as exception of the rule).

6 Discussion

Current online forums allow the user to reply to some previous post. An unthreaded forum will allow users only to overtly indicate a reply to the main posting, not a reply to some previous post, though the user can, in their comment, indicate that their reply refers to some previous poster's comment. In this regard, the unthreaded forum is homogeneous, the structure of the comments being *flat*. In a multi-threaded forum, users are able to overtly indicate a reply to the main posting as well as to previous posting; the interchanges take on a branching tree structure, a lower branch directly commenting on the immediately superior branch. However, the nature of the comment is not overtly indicated; the user may agree, disagree, or even introduce some additional statement that bears no relationship to the previous statement.

In terms of content, the current online forums are relatively unstructured and unconstrained in the sense that users could enter in arbitrary words, phrases, ungrammatical sentences, etc. The only constraint is introduced by the users themselves, which is the effort to maintain some semlance of rational organization to the course of discussion. Even if users abide by the social constraint to express themselves clearly and grammatically, there are no constraints on what post responds to what other post, the relationships among the posts, the manner of introduction of a post topic, etc. A multi-threaded forum is more structured than a forum with a homogeneous structure since

each post indicates some relationship, namely a reply to, a previous post. In general, forums have little structure and no *protocol*, which determines what reply moves a user may make.

More recent developments such as Encouraged Commentary [11] support the addition of statements into web-based text that characterise the relationship to some extent; it is similar to the comment facility in Word, where users can graphically indicate what is being commented upon. However, there is no semantics within or between statements.

The outline of the initial system above would require a range of additional functionalities:

- Add questions.
- Add redundancy check.
- Add circularity check.
- Associate the statement inputs to a parser and semantic interpreter.
- Relate the discussion to an ontology.

References

1. Wyner, A., Bench-Capon, T., Dunne, P.: Instantiating knowledge bases in abstract argumentation frameworks. In Parsons, S., ed.: Proceedings of the AAAI Fall Symposium 2009: The Uses of Computational Argumentation, AAAI Press (2009) Forthcoming
2. Reed, C., Rowe, G.: Araucaria: Software for argument analysis, diagramming and representation. International Journal on Artificial Intelligence Tools **13**(4) (2004) 961–980
3. Gordon, T.F., Prakken, H., Walton, D.: The carneades model of argument and burden of proof. Volume 171. (2007) 875–896
4. Welling, L.: PHP and MySQL web development. Pearson Education Inc. (2009)
5. Prakken, H.: Formalising ordinary legal disputes: a case study. Artificial Intelligence and Law **16** (2008) 333–359
6. Thelwall, M., Hasler, L.: Blog search engines. Online Information Review **31**(4) (2007) 467–479
7. Prabowo, R., Thelwall, M.: Finding and tracking subjects within an ongoing debate. Journal of Informetrics **2**(2) (2008) 107–127
8. Prabowo, R., Thelwall, M., Hellsten, I., Scharnhorst, A.: Evolving debate in online communication: A graph analytical approach. Internet Research **18**(5) (2008) 520–540
9. Prabowo, R., Thelwall, M.: Sentiment analysis: A combined approach. Journal of Informetrics **3**(1) (2009) 143–157
10. Wyner, A., Mochales-Palau, R., Moens, M.F., Milward, D.: Approaches to text mining arguments from legal cases. In Montemagni, S., Tiscornia, D., Francesconi, E., eds.: Semantic Processing of Legal Texts. Lecture Notes in Computer Science. Springer (2009) To Appear.
11. Wells, S., Gourlay, C., Reed, C.: Argument blogging. Technical report, University of Dundee (2009)