ARGUER: Using Argument Schemas for Argument Detection and Rebuttal in Dialogs

A.C. Restificart, S.S. Ali, and S.W. McRoy

1 Electrical Engineering and Computer Science
2 Mathematical Sciences
University of Wisconsin-Milwaukee, Milwaukee, WI, USA

Abstract

This paper presents a computational method for argumentation on the basis of a declarative characterization of the structure of arguments. The method can be used to implement a computational agent that is both able to detect arguments and to generate candidate arguments for rebuttal. The method makes no a priori assumptions about attack and support relations between propositions that are advanced by the agents participating in a dialog. Rather, using the method, these relations are dynamically established while the dialog is taking place. This allows incremental processing since the system need only consider the current utterance advanced by the dialog participant, along with the prior context, to be able to continue processing.

1 Introduction and Motivation

Argument detection is an important task in building an intelligent system that can understand and engage in an argument. An intelligent dialog system (IDS) (Bordegoni et al. (1997)) is an interactive system that tailors its responses according to the user's needs and intentions. In an IDS, it is necessary to detect whether an utterance given by the user is an argument against an utterance advanced by the system, because two agents, e.g. the system and the user, may not always agree. Each of them may attempt to resolve issues either by attacking an agent's claim or by defending its position. Thus, an IDS must be able to determine whether a proposition advanced by an agent in a dialog attacks a claim currently held by the other agent, supports it, or does neither. An IDS must also be able to generate rebuttals (utterances that attack or support previous utterances). Finally, an IDS must be able to process arguments incrementally, while the dialog is taking place. This work extends our prior work on detecting and correcting misunderstandings during the course of a dialog (McRoy and Hirst (1995), McRoy (1995), and McRoy (1998)).

The method that we describe here, which is used in our system ARGUER, uses argument schemata that match the deep meaning representation of propositions that have been advanced in a dialog. In contrast to Birnbaum et al. (1980), we present a general computational method of establishing relations between propositions. Argument schemata characterize important patterns of argument that are used to establish whether propositions support or attack other propositions. These patterns are instantiated by propositions expressed by the agents during a dialog, as well as related beliefs that the agents might hold. To account for disagreements, separate models of the agents' beliefs are maintained, both for the system and the user. Hence, a proposition believed by the system might not necessarily be believed by the user. To generate a correct and convincing response, the system considers both its own beliefs and those beliefs held by the user. In addition to allowing for incremental processing of arguments, this method is symmetric because it can be used for interpretation or generation of arguments. This is important because the system can have the role of observer or participant.

This work was supported by the National Science Foundation, under grants IRI-9701617 and IRI-9523666 and by a gift from Intel Corporation.
2 Argument Detection and Rebuttal

The architecture of ARGUER is depicted in Figure 1. The system has access to the Domain Knowledge Base (KB) (domain knowledge), Argumentation KB (argumentation knowledge, i.e. knowledge about the structure of arguments), and the Commonsense KB. (The underlying knowledge representation used by ARGUER is SNePS Shapiro and Rapaport (1992), which provides facilities for reasoning, truth-maintenance and knowledge partitioning.) The system is presumed to be the domain expert. During turn-taking (the dotted boxes in Figure 1 represent the extensions of the knowledge base used for each turn), the system has access to the users' belief models (because it is the system's models of the users). Depending on whose turn it is, the system attempts to interpret whether the current utterance attacks or supports previous claims (which may be any prior utterance).

When the user inputs an utterance, the system will attempt to interpret the user's utterance as an attack or support on a prior utterance of the system. It does so by asking, What does the user's utterance attack? and second, What does the user's utterance support? All reasoning to answer these questions occurs in an extension of the user's belief model that includes the argumentation and commonsense knowledge (which are presumed to be shared knowledge).

When the system generates a reply, it will attempt to attack or support the previous utterances of the user. Otherwise, the system will attempt to provide utterances that support its prior utterances. To generate a response, the system will reason with an extension of its knowledge including domain, argumentation, and commonsense knowledge, along with a relevant set of the user's beliefs (Ali et al. (1999)). (The latter is determined by considering applicable argument schemata).

Figure 2 is an example dialogue that includes an argument. In the figure, S1 and S2 are utterances of the system. U1 is the user's utterance. To detect that U1 attacks S1, the system makes use of the argument schema rule: If X is an utterance implying Y, then NOT Y is an attack to X (which is in the Argumentation KB). A rule in the commonsense KB allows the system to derive that requiring something of an agent (which follows from the imperative form of S1) implies that there is a need for that thing by the agent. S1 implies Y ("There is a need for a blood pressure check.") Thus, using the above argument schema rule, (NOT Y) ("there is no need for a blood pressure check") is an attack to X ("system requires the user to have a blood pressure check").

This argument schema can also be used to generate a rebuttal. Suppose the user said S1. Using the argument schema rule describe above, Y is instantiated as "there is a need for a blood pressure check". This, in turn, allows the system to select (NOT Y) as an attack to X to make the utterance U1 to rebut S1.