An Multi Agent System for Argumentation
Research Report

Alexandru Sorici, Alin Danciu, and Tudor Berariu

Faculty of Automatic Control and Computer Science,
University "Politehnica" of Bucharest, Romania

Abstract. Key words: natural language processing, machine learning, argumentation, statistical learning, multi agent system

0.1 Argumentation in Computer Science

With regards to computer science, the study of argumentation is crucial in many artificial intelligence and natural language research problems.

For example, in the field of Multi Agent Systems (MAS) reasoning agents need to communicate with each other and apply argumentation-based reasoning mechanisms to resolve the conflicts arising from their different views of goals, beliefs, and actions. Another example are question answering systems, which deal with finding the correct response to questions like “Why was this decision taken?” and therefore integrate the analysis of argumentation as a crucial part of identifying the answer to the questions as well as the pros and cons that make up the answer.

The field of computer-supported collaborative learning (CSCL) has, in particular, been interested in argumentation and how students can benefit from it (Stegmann et al. 2007 - *Facilitating argumentative knowledge construction with computer-supported collaboration scripts*;). So-called “collaborative argumentation” is viewed as a key way in which students can learn critical thinking, elaboration, and reasoning. Therefore, it is a crucial point to understand the characteristics and models of argumentation.

Argumentation mining, the subject we are trying to approach, is a new research area that moves between natural language processing, argumentation theory and information retrieval. The aim of argumentation mining is to automatically detect the argumentation of a document and its structure.

Some relevant work in this area has been done by R.M. Palau and M-F Moens in *Argumentation mining: the detection, classification and structure of arguments in text*. In the paper they analyze the main research questions when dealing with argumentation mining and the different methods they have studied and developed in order to successfully confront the challenges of argumentation mining in legal texts. A more structured approach is taken by Safia ABBAS , Hajime SAWAMURA in *ALES: An Innovative Argument Learning Environment*. The environment uses different mining techniques to manage a highly structured arguments repository.

Overall, the task of argument extraction from free text is a new and difficult research area, one that will hopefully see a lot of activity in the near future, because the results that can be obtained will be of great use in many related fields of natural language processing.

1 Argumentation Concepts Overview

A nice overview on argumentation tasks and concepts was given by Douglas Walton in his research reports. The next sections are inspired by his work.

1.1 Main problems

There are four tasks undertaken by argumentation:

identification which means finding the premises and conclusion fo arguments in a text and fitting that argument in an argumentation scheme;

analysis which involves the identification of implicit premises or conclusion of arguments (such an argument is called an enthymeme);

evaluation where the strength of an argument must be determined using a general criteria;

invention which means constructing new arguments that can be used to prove some conclusion

1.2 Argument definition

An argument is a set of statements (propositions), made up of three parts, a conclusion, a set of premises, and an inference from the premises to the conclusion. An argument can be supported by other arguments, or it can be attacked by other arguments, and by raising critical questions about it.

The definition of 'argument' relied on so far could be called a minimal inferential definition, and the method of argument diagramming shown so far fits this minimal definition. The boxes represent propositions and the arrows represent inferences from some propositions to others.

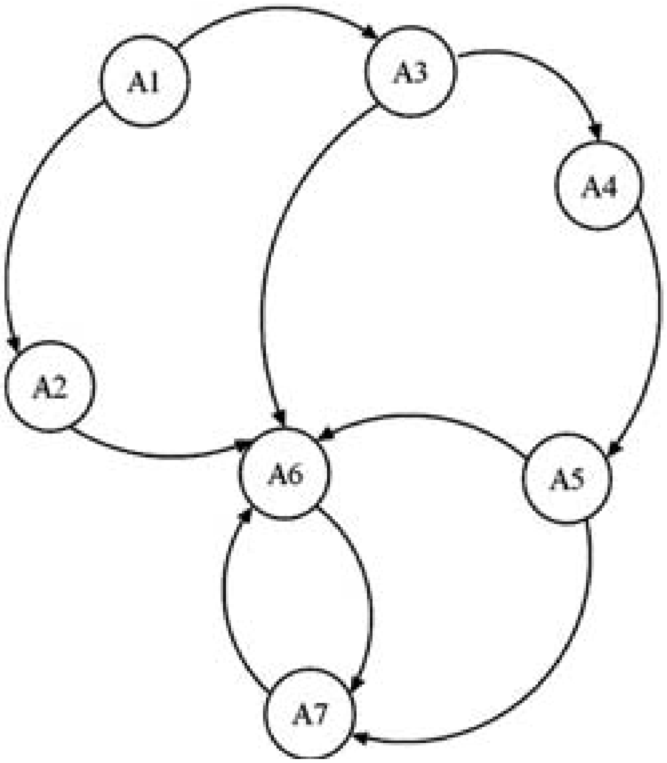
The general approach or methodology of argumentation can be described as distinctively different from the traditional approach based on deductive logic. The traditional approach concentrated on a single inference, where the premises and conclusion are designated in advance, and applied formal models like propositional calculus and quantification theory determine whether the conclusion conclusively follows from the premises. This approach is often called monological.

In contrast, the argumentation approach is called dialogical (or dialectical) in that it looks at two sides of an argument, the pro and the contra. According to this approach, the method of evaluation is to examine how the strongest arguments for and against a particular proposition at issue interact with each other, and in particular how each argument is subject to probing critical questioning that reveals doubts about it. By this dialog process of pitting the one argument against the other, the weaknesses in each argument are revealed, and it is shown which of the two arguments is the stronger.

1.3 Argument Attack and Refutation

There are several ways to attack an argument and the easiest way is to ask an appropriate critical question that raises doubt about the acceptability of the argument. When this happens, the argument temporarily defaults until the proponent can respond appropriately to the critical question. Another way to attack an argument is to question one of the premises. A third way to attack an argument is to put forward counter-argument that opposes the original argument, meaning that the conclusion of the opposing argument is the opposite (negation) of the conclusion of the original argument.

A simple way to represent a sequence of argumentation is using a directed graph:



1.4 Argumentation Schemes

Some of the most common schemes are: argument from witness testimony, argument from expert opinion, argument from popular opinion, argument from example, argument from analogy, practical reasoning (from goal to action), argument from verbal classification, argument from sign, argument from sunk costs, argument from appearance, argument from ignorance, argument from cause to effect, abductive reasoning, argument from consequences, argument from alternatives, argument from pity, argument from commitment, ad hominem argument, argument from bias, slippery slope argument, and argument from precedent. Each scheme has a set of critical questions matching the scheme and such a set represents standard ways of critically probing into an argument to find aspects of it that are open criticism.

1.5 Types of dialog

There are six types of dialogs:

Persuasion generated by a **Conflict of Opinions** in which each party tries to **Persuade Other Party** in order to **Resolve or Clarify Issue**

Inquiry generated by the **Need to Have Proof** in which each party tries to **Find and Verify Evidence** in order to **Prove (Disprove) the Hypothesis**

Negotiation generated by a **Conflict of Interests** in which each party tries to **Get What He Most Wants** in order to reach a **Reasonable Settlement Both Can Live With**

Information-Seeking catalyzed by the **Need of Information** in which each party **Acquires or Gives Information** in order to **Exchange Information**

Deliberation generated by a **Deliberation Dilemma or Practical Choice** that is needed in which the participants try to **Co-ordinate Goals and Actions** in order to **Decide on Best Available Course of Action**

Eristic generated by a **Personal Conflict** in which each party tries to **Verbally Hit Out at Opponent** in order to **Reveal Deeper Basis of Conflict**.

On our project we will probably focus on inquiry dialogs.

2 NLP Aspects Overview

As stated in the last section the goal of project becomes that of identifying the arguments and the relations that exist between them in a given text. As such, our task can be modeled as a classification problem.

As presented previously, there exist a number of models for both the intrinsic structure of a model, as well as for the relations that can hold between them. The above sections gave a classification of the different types of dialog and argumentation schemes. What is interesting to notice is that each type of dialog supports some argumentation schemes better than others. Thus, having chosen one specific type of dialog, the most relevant argumentation schemes pertaining to that dialog model will be chosen as structure guidelines for our arguments. In addition, a Toulmin (S. E. Toulmin. *The Uses of Argument*) based alternative representation of an argument will be considered.

2.1 Main problems

Having laid out this model, we are faced with the following challenges of Argumentation Mining:

Detect all the arguments in a free text (classification problem)
Determine argument limits (segmentation problem)
Determine arguments type (complex)
Detect argumentation structure (complex)

The Corpus that will be used to for solving the above problems is the one provided by *Arauacaria* (*http://www.arg.dundee.ac.uk/projects/araucariadb/search.php*), which, albeit being not very large, has the advantage of giving annotated argument examples.

In the work by R.M. Palau and M-F Moens: *Argumentation mining: the detection, classifcation and structure of arguments in text* some solutions to this problems are proposed, some of which we consider to be very useful for our project. For the problem of argument detection statistical classifiers are employed. In particular, good results can be obtained by using the *maximum entropy model* and *multinomial naive Bayes* classifier which learns a model of the joint probability of an element x and its label y, $p(x, y)$, and makes its predictions by using Bayes rule to calculate p(y|x) and then selects the most likely label y.

For the problem of determining argument limits (grouping of argumentative statements into their corresponding argument) we consider computing the semantic distance between different argumentative units (sentences) and group sentences in one argument if they discuss content that is semantically related. Like in the work of Palau and Moens, we assume that the relatedness of two sentences is a function of the relatedness of their words. We want to employ an ontology based semantic relatedness measure, where the relatedness of words depends on their semantic distances in a lexico-semantic resource such as WordNet.

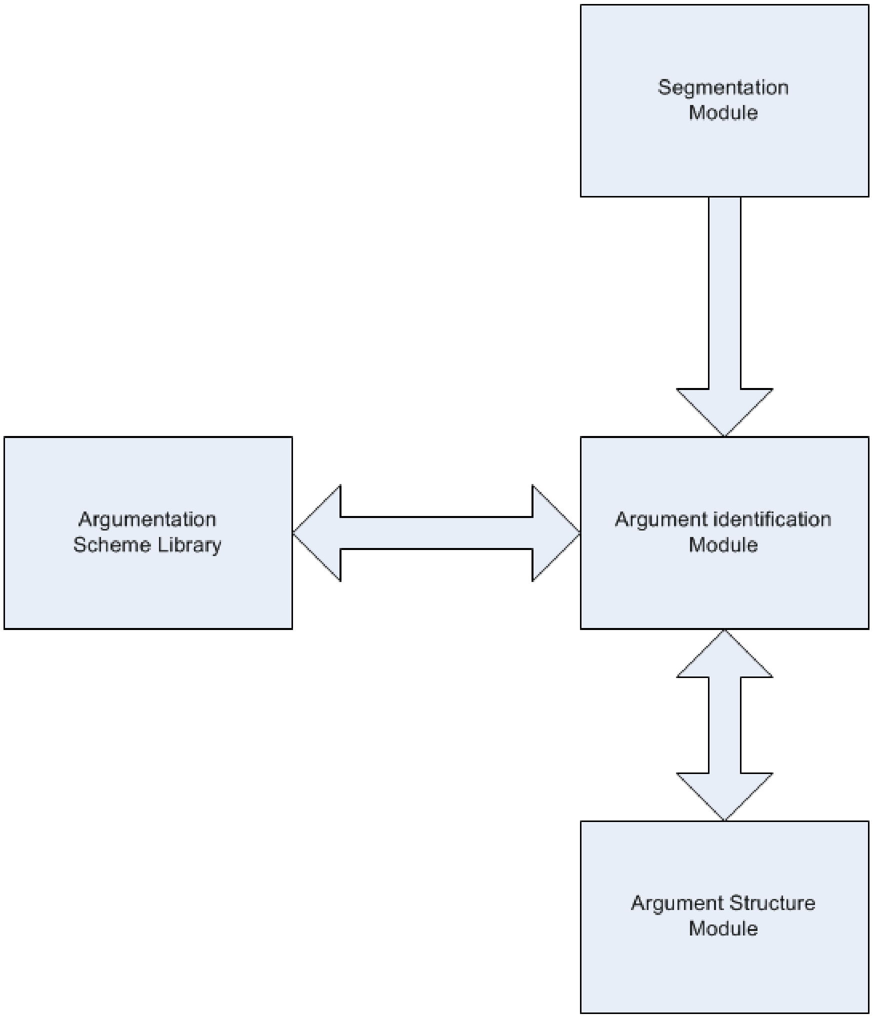
In the determination of argument types, we also consider the use of statistical classifiers. The problem here is that while a classification of argumentative statements into premises and conclusions is possible (as shown by Palau and Moens by use of a SVM), the task of determining a specific scheme for an argument is much more challenging. In particular, we have to rely on some domain ontology in order to be able to find semantic features that will allow us to classify the arguments as belonging to a certain scheme.

Extensive use of other NLP tools such as POS taggers will also be made in order to enrich the possible set of features used for classification.

Lastly, the problem of determining the argumentation structure, that is, of relations existing between the arguments, will have to take all the results obtained by the previous steps into consideration. The employed algorithm will make the assumption that the discourse follows the rules of the chosen dialog type and that arguments used within it belong to the appropriate argumentation schemes.

3 Argumentation System architecture - Module description

This section gives a general overview of the most important modules of our application. We will give a graphical representation of the main modules and of the relation between them. Afterwards, we will give a more detailed description of each module, highlighting the approaches and algorithms considered for their implementation.


Segmentation Module
Argumentation Scheme Library
Argument identification Module
Argument Structure Module

3.1 Argument Identification Module

The Argument Identification Module classifies all sentences into argumentative and non-argumentative utterances. This modules essential role is to determine which parts of the text are also parts of argument structures.

This module will use a classifier in order to extract argumentative. We will take, most probably, one of the next two options for implementation: a naive Bayes classifier or a maximum-entropy method.

The most important features our classifier will use are:

Unigrams Each word in the sentence.
Bigrams Each pair of successive words.
Trigrams Each three successive words.

Adverbs Detected with a part-of-speech (POS) tagger (e.g. QTag 1).

Verbs Detected with a POS tagger. Only the main verbs (excluding "to be", "to do" and "to have") are considered.

Modal auxiliary Indicates if a modal auxiliary is present using a POS tagger.

Word couples All possible combinations of two words in the sentence are considered.

Text statistics Sentence length, average word length and number of punctuation marks.

Punctuation The sequence of punctuation marks present in the sentence is used as a feature (e.g. ":."). When a punctuation mark occurs more than once in a row, it is considered the same pattern (e.g. two or more successive commas both result in ",+").

Key words Keywords refer to several words or word sequences obtained from a list of terms indicative for argumentation. Examples from the list are "but", "consequently", and "because of".

3.2 Segmentation Module

The Argumentation Identification Module uses this module to partition different argumentative sentences (units) into their corresponding arguments, that is, to determine the limits of each individual argument.

In order to achieve this we opt to calculate the semantic distance between the different argumentative units, and group sentences in one argument if they discuss content that is semantically related. As in the work of Palau and Moens, we assume that the relatedness of two sentences is a function of the relatedness of their words.

Given this assumption we take the semantic relatedness of words to be given by their semantic distances in a lexico-semantic resource such as WordNet.

Measures of relatedness are more general in that they can be made across part of speech boundaries, and they are not limited to considering is-a relations.

Several measures of relatedness which are based on the WordNet database can be used.

- The hso (Hirst and St-Onge, 1998) measures classifies relations in WordNet as having direction, and then establishes the relatedness between two concepts A and B by finding a path that is neither too long nor that changes direction too often.
- The lesk (Banerjee and Pedersen, 2003) and vector measures incorporate information from WordNet glosses. The lesk measure finds overlaps between the glosses of concepts A and B, as well as concepts that are directly linked to A and B.
- The vector measure (Patwardhan, 2003) creates a cooccurrence matrix for each word used in the WordNet glosses from a given corpus, and then represents each gloss/concept with a vector that is the average of these cooccurrence vectors.

WordNet provides modules that implement these measures and that offer a specific API for their usage.

Using these measures we can compute an overall semantic relatedness of the argumentation units. By means of a clustering algorithm that will use the distances given by the above mentioned methods we can then segment sentences into their corresponding arguments (clusters), thus finding the intended argument limits.

3.3 Argumentation Scheme Library

The Argumentation Scheme Library contains abstract forms of argument that capture patterns of reasoning. Each argumentation scheme describes the relations between internal parts (the premises and the conclusion) of a complex argument and, also, the set of critical questions that may attack that argument. Chris Reed and Douglas Walton, in their research report "Towards a Formal and Implemented Model of Argumentation Schemes in Agent Communication" describe the conventional techniques to handle the structure of argumentation schemes in such way that agents may use them in reasoning and also that communication structures can be built around those schemes.

Example of Argumentation Scheme Argument from Position To Know:

Major Premise : Source a is in a position to know about things in a certain subject domain S containing proposition A.
Minor Premise : a asserts that A (in Domain S) is true (false).
Conclusion : A is true (false).

This type of argument is defeasible by questioning. Matching the argument from position to know are three critical questions:

CQ1 : Is a in a position to know whether A is true (false)?
CQ2 : Is a an honest (trustworthy, reliable) source?
CQ3 : Did a assert that A is true (false)?

Our Argumentation Scheme Library will be trained from Araucaria DB, which contains several real instantiations of argumentation schemes. The representation used is Argument Markup Language, based upon the industry standard XML.

3.4 Argument Structure Module

The Argument Identification Module and Segmentation Module were used to distinguish between argumentative and non-argumentative sentences and to group argumentative sentences together into their corresponding argument.

The purpose of this module thus becomes to determine the internal structure of an argument, i.e. the organization of its clauses into conclusions and supporting premises.

We will consider two possible approaches for this task.

The first relies again on work done by Palau and Moens in Argumentation Mining: The Detection, Classification and Structuring of Arguments in Text and considers the partition of an arguments units into premises or conclusions.

For each argument, certain features will be determined. The list can comprise features such as:

Tense of Main Verb : Tense of the verb from the main clause of the sentence; having as nominal values “Present”, “Past” or “No Verb”.

History : The most probable argumentative category of previous and next sentence.

Rhetorical Patterns : Type of rhetorical pattern occurring on current, previous and next sentences (e.g. “however,”)

Argumentative Patterns : Type of argumentative pattern occurring in sentence

A SVM (Support Vector Machine) based algorithm that uses these features will then be used to classify the sentences of an argument into either premises or conclusions.

The second approach is an extension of the first one. Rather than just distinguishing between premise and conclusion, this approach tries to map the arguments structure onto one or several argumentation schemes.

The definition and format of these schemes is provided by the Argumentation Scheme Library module.

Making use of more extended features, that will also include the ones listed above, our SVM based algorithm will this time be trained to distinguish between different argumentation schemes.

The role of the segmentation module is that to partition different argumentative sentences (units) into their corresponding arguments, that is, to determine the limits of each individual argument.

As presented in the description of the NLP module capabilities, we opt to calculate the semantic distance between the different argumentative units, and group sentences in one argument if they discuss content that is semantically related.

The module receives a list of argumentative sentences as its input. It then uses a word tokenizer to split the sentences into lists of words. Stopwords such as ”the”, ”in”, ”with” etc. are removed before going to the next step.

We assume the semantic relatedness of words to be given by their semantic distances in a lexico-semantic resource such as WordNet. We use the lin similarity measure to compute the relatedness of two synsets. As each word in a sentence can have several senses, one would first have to go through a word sense disambiguation process to determine the most probable meaning. As this process is time-consuming, the current approach employs the use of the most common sense, as defined by WordNet, for each word we encounter.

Using the lin similarity we compute a word similarity matrix between every pair of words, one from each sentence.

To compute the estimated similarity between two sentences we then use the following method. Given two sentences A and B, for each word from sentence A we compute the most similar word from sentence B, as given by the previously computed matrix.

$$b^* = arg \max_b Sim(a, b)$$

In a similar manner, for each word in sentence B we compute the most similar word from sentence A.

$$a^* = arg \max_a Sim(a, b)$$

Afterwards the similarity between sentences A and B is given by

$$(\sum a^* + \sum b^*)/(2 * (\mid A \mid + \mid B \mid)$$

The above process yields a sentence similarity matrix. This matrix is used as an input to a hierarchical clustering algorithm. We take an empirically determined cutoff distance of 0.875 to cut the resulting cluster dendrogram at the corresponding depth. This results in a list of sentence clusters, each of which contains the components of an individual argument.

Here is a simple example. The initial text:

“It is well established that if any statement is made on the floor of the House by a Member or Minister which another Member believes to be untrue, incomplete or incorrect, it does not constitute a breach of privilege. In order to constitute a breach of privilege or contempt of the House, it has to be proved that the statement was not only wrong or misleading but it was made deliberately to mislead the House. A breach of privilege can arise only when the Member or the Minister makes a false statement or an incorrect statement willfully, deliberately and knowingly. On a perusal of the comments of the Ministers in the matter, I am satisfied that there has been no misleading of the House by them as alleged by the Member. I have accordingly disallowed the notice of question of privilege. Copies of the comments of the Ministers have already been made available to Dr. Raghuvansh Prasad Singh.”

The determined word lists:

L1: [u‘It’, u‘well’, u’established’, u’statement’, u’made’, u’floor’, u’House’, u’Member’, u’Minister’,u’another’, u’Member’, u’believes’, u’untrue’, u’incomplete’, u’incorrect’, u’constitute’, u’breach’, u’privilege’]

L2: [u’In’, u’order’, u’constitute’, u’breach’, u’privilege’, u’contempt’, u’House’, u’proved’, u’statement’, u’wrong’, u’misleading’, u’made’, u’deliberately’, u’mislead’, u’House’]

L3: [u’A’, u’breach’, u’privilege’, u’arise’, u’Member’, u’Minister’, u’makes’, u’false’, u’statement’, u’incorrect’, u’statement’, u’wilfully’, u’deliberately’, u’knowingly’]
L4: [u’On’, u’perusal’, u’comments’, u’Ministers’, u’matter’, u’I’, u’satisfied’, u’misleading’, u’House’, u’alleged’, u’Member’]
L5: [u’I’, u’accordingly’, u’disallowed’, u’notice’, u’question’, u’privilege’]
L6: [u’Copies’, u’comments’, u’Ministers’, u’already’, u’made’, u’available’, u’Dr’, u’Raghuvansh’, u’Prasad’, u’Singh’]

The (simmetric) sentence similarity matrix:

	1	*2*	*3*	*4*	*5*	*6*
1	1.00	0.24	0.24	0.19	0.10	0.13
2	0.24	1.00	0.21	0.19	0.14	0.10
3	0.24	0.21	1.00	0.16	0.13	0.11
4	0.19	0.19	0.16	1.00	0.13	0.12
5	0.10	0.14	0.13	0.13	1.00	0.06
6	0.13	0.10	0.11	0.12	0.06	1.00

The resulting sentence clustering (only the indexes of the sentences are show):

$$[[5],[4,3,1,0,2]]$$

We can see that sentences 1-5 belong to one argument and sentence 6, which is actually non-argumentative, this being only an illustrative example, belongs to a separate cluster.

The current method yields some promising results, but there is still room for improvement, especially in the process of determining word similarities, where we hope to be able to implement a more advanced measurement solution. In particular, the Lin measure can only compute the similarity between words which have the same part of speech (noun, verb etc). The lesk or gloss-vector measures, which we will try to implement in future developments can cross these boundaries and are not limited by is-a relations.