State of the ART: An Argument Reconstruction Tool

Winkels, R.; Douw, J.; Veldhoen, S.

Published in:
Semantic Processing of Legal Texts: SPLeT-2014: workshop programme

State of the ART: an Argument Reconstruction Tool

Radboud Winkels, Jochem Douw, Sara Veldhoen
Leibniz Center for Law,
University of Amsterdam
winkels@uva.nl

Abstract

This paper describes the outcomes of a series of experiments in automated support for users that try to find and analyse arguments in natural language texts in the context of the FP7 project IMPACT. Manual extraction of arguments is a non-trivial task and requires extensive training and expertise. We investigated several possibilities to support this process by using natural language processing (NLP), from classifying pieces of text as either argumentative or non-argumentative to clustering answers to policy green paper questions in the hope that these clusters would contain similar arguments. Results are diverse, but also show that we cannot come a long way without an extensive pre-tagged corpus.

Keywords: argument mining, clustering, cluster tendency, policy modelling, machine learning

1. Introduction

Before publishing a policy white paper, the European Union often publishes a draft, a green paper, to stimulate discussion and enable public consultation. The green paper provides the opportunity to companies and individuals to respond to the draft and provide arguments in favour or against it. Typically such a green paper raises issues and ask questions like "Should there be encouragement or guidelines for contractual arrangements between right holders and users for the implementation of copyright exceptions?".[1] Exploring and indexing these replies and their arguments from external sources is difficult and time consuming. The goal of EU FP7 project IMPACT ("Integrated Method for Policy Making Using Argument Modelling and Computer Assisted Text Analysis") was to provide means to support this process.[2] This includes a so-called "Argument Reconstruction Tool" (ART) that enables users to easily copy and store text fragments and relate them using formal argument structures. Part of the foreseen functionality of the tool was to help the user by finding text fragments that contain arguments and possibly suggesting argument schemes that are used.
This paper introduces the ART and focusses on two experiments in automated argument finding and reconstructing.

2. The ART

The ART is implemented as a Rich Internet Application (RIA). Arguments are stored using a separate storage class that abstracts away from the current MySQL implementation. Users can copy and paste any piece of text into the system by hand and construct arguments at different levels of detail:

Unary Relations (UR) We are enabling users to start with annotating texts with qualifications like "there is an argument somewhere here" or "this is a proposition that is part of an argument". These are unary relations on the pieces of text, usually consisting of one or more sentences.

Binary Relations (BR) In addition to that we enable users to make binary relations between arguments or parts of arguments. These binary relations can e.g. be of the form "A supports B" or "A attacks B". These relationships can actually exist on several different levels: it can e.g. be a relation between two entire arguments (represented in either of the three states below) or between two variables (necessitating the argument to be modelled at the PCLAS level).

Abstract Argument Scheme (AAS) This relationship connects one or more premises to a conclusion.

Proposition Level Argument Scheme (PPLAS) We make a distinction between different sorts of premises based on an argument scheme. For the Argument from Credible Source (ACS) scheme, we could make a distinction between the atomic terms "Newton was an expert in science", "Newton said that things always fall down" and "Statements about things falling down fall within the domain of science". These statements have three types, that could be called respectively "Credible source assumption", "Person asserts statement", and "Asserted statement within domain".

Predicate Level Argument Scheme (PCLAS) This is the finest level of argumentation representation. When we take the ACS scheme as example again, we make a distinction between atomic statement types "expert", "statement" and "domain". These have a fixed meaning within the ACS scheme, but can also be coupled as predicates by saying asserts(*expert*, *statement*), or at the instantiated level asserts("Einstein", "All things fall down").

All these schemes can be either uninstantiated or instantiated.
The ART currently has three basic argumentation schemes:

1. General Argument Scheme: The most simple one, just consisting of one or more premises and a conclusion.

[1] From "Copyright in the Knowledge Economy".

[2] See `http://www.policy-impact.eu/` for more information.

2. Credible Source Argument Scheme: It consists of a proposition from a certain domain stated by a particular source. See figure 1.
3. Practical Reasoning Argument Scheme: Consists of an action proposed by an agent in particular circumstances described by one or more propositions, leading to consequences described by one or more propositions to promote one or more values.

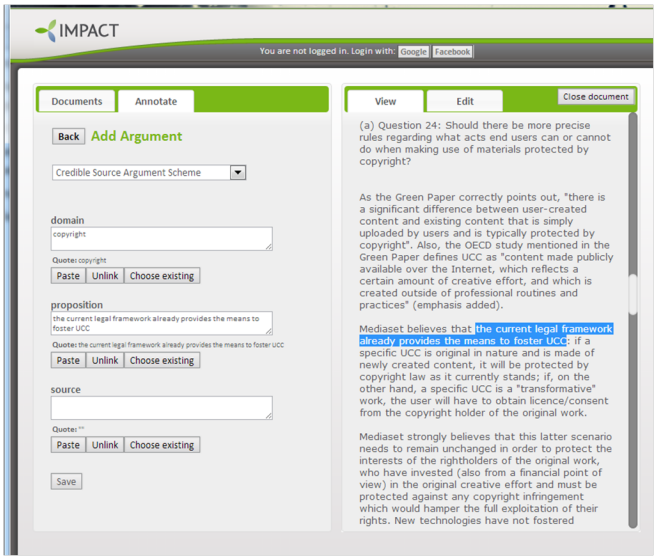

Figure 1: *A partially filled credible source argument scheme.*

3. Extraction of Arguments

Manual extraction of arguments from a text is a non-trivial task. In (Mochales and Moens, 2011), an example is given of three annotators that had to identify arguments in verdicts of the ECHR.[3] They write: *"The overall process took more than a year and included three annotators and one judge to solve disagreements. Once the task was completed, the annotation obtained a 75% agreement between annotators [...]"* . It would be helpful if the machine could detect the use of arguments and suggest schemes and perhaps even prefill them and present them for verification to the human users.

3.1. Related Research

In the ART, arguments can be extracted manually by users. We have the ambition to employ natural language processing (NLP) to recognise the arguments inside a natural language text such as a green paper, a website or a blog. In general one can state that up to the beginning of the IMPACT project in 2010, hardly any research had been devoted to automated argument reconstruction from natural language texts (cf. (Moens et al., 2007), (Palau and Moens, 2009)).

Brüninghaus and Ashley (Brüninghaus and Ashley, 2005) built systems that recognise relevant factors in legal texts and then proceed to generate (and evaluate) an argumentation from those facts. Classifiers were made to determine if a certain factor appeared in each sentence. These factors came from a list of factors from trade secret cases, and are more specific than the arguments that a generic tool should recognise. Different machine learning approaches were tested to train these classifiers, with three different forms of data representation. TiMBL worked best with data represented as propositional patterns (F-measure of 0.26).

An actual attempt at argument detection has been made by Mochales Palau and Moens (Palau and Moens, 2009). They perform three steps: 1. classification of a proposition as argumentative or non-argumentative; 2. classification of an argumentative proposition as a premise or a conclusion; 3. detecting the argument structure.

In a corpus based on diverse sources (the so-called structured Araucaria corpus consisting of 641 arguments from newspaper articles, online discussion boards, and magazines) they were able to detect arguments with 73% accuracy; classify premises and conclusions with a F1 measure of about 70%, and detect argumentation structures with about 60% accuracy. The argument structure is detected using a context-free grammar. The classification was attempted with both machine learning classifiers and context-free grammars, with the machine learning classifiers (maximum entropy model and support vector machines) leading to the best results.

A somewhat different approach is to start with a classification of the relation between two text fragments rather than the classification of the text fragments themselves. Marcu and Echihabi (Marcu and Echihabi, 2002) focus on the automated recognition of discourse relations, which are descriptions of how two spans of texts relate to each other. They created a corpus containing different text fragments and the relation between them, confining themselves to the relations contrast, cause explanation-evidence, elaboration and condition. They then used Naive Bayes classifiers to distinguish between two relations, which had a performance of between 64% and 93%, depending on the relations that were compared.

These approaches suggest that a machine learning approach will be better for the task of detecting arguments than a pattern-based approach, but that identifying relevant patterns is still valuable, as they can be included as features for the machine learning approach.

4. First Experiment

As explained above, literature suggests the use of machine learning techniques. However, the dataset required to train such machine learning techniques will be developed using the ART tool once it is operational. Unfortunately we were not able to accumulate a large enough dataset from other sources, so we resorted to keyword-based tagging based on manual inspection of sources.

The domain consists of replies to the EU green paper "Consultation on the Commission Report on the enforcement of intellectual property rights".[4] These documents are mostly written in a neutral style, with a low amount of sentiment cues. The arguments provided often consist of just propositions without keywords indicating their role or the fact that

[3] The European Court of Human Rights in Strasbourg, France.

[4] The replies can be found at : `http://ec.europa.eu/`

it is an argument at all. Domain knowledge and common sense are required to reconstruct the argumentation in these responses. Finally, almost every argument is an implicit "argument from position to know" (Walton, 2002). This is inherent to the context of green paper discussions, which is that companies and organisations establish themselves as being in the position to know about the topic at hand and then try to convince the EU of a particular standpoint.

4.1. Keywords and Regular Expressions

The first step was to see if the documents contained any keywords that indicate the use of argumentation. The following documents were used as training set.

Source	Total words
ANBPPI/BNVBIE	5165
Google	4830
Bits of Freedom	2150
Ericsson	1919
Business Europe	1068

Three observations can be made. (1) The frequency of most keywords, if not all, is very low (a small portion is shown in table 1). The documents contain arguments in nearly every paragraph, but only a small portion of these arguments uses identifiable keywords. (2) The use of argumentative expressions, linguistic constructions and vocabulary differs dramatically *over* documents, but is rather consistent *within* a document. This is one of the reasons for the overall low frequencies of keywords. (3) The keywords that were useful can be divided in roughly three categories: *Structure segments* that indicate structural relations between sentences (e.g. for example, firstly); *Argumentation segments* that indicate argumentational relations between (parts of) sentences (e.g. concludes, therefore, in contrast with, see table 1); and *Sentiment segments* that are not directly linked to argumentations but do indicate the expression of an opinion which can indirectly indicate that an argumentation is used (e.g. essential, believe). For more extensive research see (Knott and Dale, 1994).

Segment	**BoF**	**Google**	**Ericsson**	**BEurope**	**ANBPPI**	**Total**
Argumentation segments						
however	1	4	3	1	7	16
thus / therefore	2	0	4	0	6	12
lead(s) to / has resulted in / result	5	0	2	1	2	10
conclude(s) / conclusion	6	1	1	0	0	8
assumption/assume	3	1	1	0	0	5
pointed out	0	4	0	0	0	4
at odds	4	0	0	0	0	4
since	1	1	1	0	1	4

Table 1: Most frequent argumentative keywords in train set.

The next step was to construct regular expressions from these keywords to tag sentences with an argumentation indication in the *test* set. Three were created: one that matches any of the keywords or combinations of them, one that indicates some sort of conclusion and one that indicates some sort of premise. As an example we present the regular expression for conclusions below:

```
(therefore|conclu(de(s|d)?|sions?)?|in fact|
thus|hence|(this|that) is why)|
(support(s|ed|ing) the conclusion|
In sum|hereby|by doing so)
```

The regular expressions were applied on the test set consisting of the following sources:

Source	Total words
PPL UK	1181
Royal TNT Post	1264

When applying the regular expression that matches any keyword on our test set the following confusion matrix was achieved:

		Manual					
		Arg	Ntrl	Ttl	Prec	Rec	F
Tagging	Argum	16	7	23	69.6%	40.0%	50.8%
	Neutral	24	65	89	73.0%	79.5%	76.1%
	Total	40	72	112	72.3%	72.3%	72.3%

About 35% of the sentences in the test set are manually tagged as argumentative; not even half of these were found using the regular expression (recall of 40%). Only 7 sentences were incorrectly classified as argumentative (few false positives). An obvious reasons for the low recall is the observed difference in language use across authors.
When applying the other two regular expressions, both recall and precision are very low for finding conclusions (F-score of 14.3%) and low for premisses (F-score of 46.8%). Although the results are in some cases quite good, there are two factors that must be taken into account. Firstly, the size of the train and test set is too small to get real representative results. Secondly, recall and f-score values are much higher for the neutral classes than the actual classes we want to find (*Argumentative*, *Conclusion* and *Premise*). Detecting *Argumentative* works better than detecting premises, which works better than conclusions, which score the worst.

5. A second Experiment

Since we do not have a tagged corpus of arguments, neither in the domain of EU green papers, nor in any other comparable domain, we decided to explore the use of unsupervised techniques. Can we find clusterings of answers to green paper questions that correlate to the use of specific types of arguments? Even if we cannot decide which argument type is exactly used, it may help policy analysts if we can provide them with clusters of similar ones.
A different EU Green Paper on "Copyright in the Knowledge Economy" contains 25 questions belonging to five distinct topics. We have used the 159 unique replies in English (from the 374 replies in total). They contain around 1300 answers to specific questions, differing in length.
In GATE[5], we created a pipeline to annotate the questions and answers in the documents after exporting them to plain text. The output of this pipeline was a set of XML documents with the annotations as in-line XML tags. We have

[5]GATE is open source software capable of solving many text processing problems, see http://gate.ac.uk/

only taken into account the answers to single questions (as opposed to general remarks and answers to a range of questions). An answer consists of one or more lines of text. The number of the question being addressed was assigned as an attribute to the XML-tag for every answer.

5.1. Clustering

We have compared a number of clustering methods. A distinction can be made between partitioning and hierarchical approaches. Partitioning cluster algorithms output a hard partition that optimizes a clustering criterion. Hierarchical algorithms produce a nested series of partitions based on a criterion for merging or splitting clusters based on similarity (Jain et al., 1999). Applying different hierarchical clustering methods did not seem to work; we mostly got one cluster containing much (>95%) of the data. Partitioning methods resulted in more equally sized clusters, so we have focused on these algorithms.

The first method we used is Expectation Maximization (EM), which assigns a probability distribution of each instance indicating the probability of it belonging to each of the clusters. This algorithm is capable of determining the number of clusters by cross validation (Moon, 1996). Another method is SimpleKMeans. It starts with a random initial partition and keeps reassigning the patterns to clusters based on the similarity between the pattern and the cluster centers (Jain et al., 1999). XMeans and FarthestFirst are extensions of the SimpleKMeans, determining the number of clusters and choosing the initial centroids to be far apart respectively. Finally we applied sIB (Sequential Information Bottleneck), which is like K-means, but the updates aren't performed in parallel (Slonim et al., 2002).

5.2. Finding Topics

First we tried a bag-of-words approach to find clusters of documents, i.e. complete answers. All answers to all questions were taken into account. The attributes source, question number and the topic of the question were added as attributes to be used for the analysis; these were not handed to the clusterer. The text content of the answers was filtered using a stop list[6].

The data was then loaded into WEKA Explorer[7] where the content attribute was converted to a series of attributes serving as a bag-of-words. The filter StringToWordVector was used, applying IDF-TF Transform and normalizeDocLength (for normalizing the values). The minTermFreq was set to 10, thus creating around 100 attributes. The outputWordCounts was set to true, creating numeric values rather than booleans. Finally, a stemming algorithm was used to map syntactically related words to the same stem.

We applied EM clustering to the data, leaving the number of clusters to be created open. The random seed was set to 100 (default). The algorithm grouped the 1301 instances into 11 clusters, with cluster sizes ranging from 39 to 266. Every instance in the dataset is an answer to a specific question, belonging to a topic. Beside, each instance has an origin, a source document. Three matching matrices were built relating the clusters to questions, sources, and topics. The latter is shown here for illustration:

Cluster → Topic ↓	0	1	2	3	4	5	6	7	8	9	10
General	49	34	9	12	6	73	1	9	60	3	16
ELA	28	40	3	114	9	54	52	17	58	7	22
EPD	12	141	3	1	20	2	2	12	27	1	38
TR	17	36	87	2	38	18	3	16	16	26	3
UCC	61	15	2	0	2	2	2	6	11	2	1

ELA = Exceptions Libraries Archives; EPD = Exceptions for People with Disability; TR = Teaching Research; UCC = User Created Content

There are many evaluation metrics available to define the extrinsic quality of a partitioning. In (Amigó et al., 2009) a wide range of metrics is analyzed according to a few intuitive constraints. The B-Cubed metric was found to be the only one satisfying all the constraints. We have used this metric to compare the clustering to the three classifications. The precision and recall are computed for each entity in the document and then combined to produce final precision and recall numbers for the entire output.

The recall, precision and F-score of the clustering compared to the three classifications are:

Classification	Precision	Recall	F-score
Question	0.123	0.309	0.176
Topic	0.420	0.219	0.288
Source	0.027	0.232	0.049

Although the first experiment showed that linguistic constructions and vocabulary differed from writer to writer, in this experiment we see that the clustering tends to correspond more to the (topics of the) questions than to the authors: compared to the other two, the scores of the 'source' classification are quite bad. There is hardly any correspondence between the author of a reply and the cluster it is assigned to. Note that in this experiment the closed-class or function words were filtered out of the text, which was not the case in the first experiment.

This finding endorses our idea of using lexical analysis to find pieces of text expressing the same ideas or subjects. However, the scores on the other two classifications are quite low as well, so it is very well possible that there is not enough information in the bag of word features to get a proper semantic grouping.

5.3. Finding Arguments

This section describes the experiments with a finer granularity. The dataset contains all answers to a specific question, the instances are the paragraphs that the answers consist of. We aim for a clustering that expresses lines of argumentation. The procedure to represent the data is the same as before except that the minTermFreq was set to 4, because the dataset is much smaller and all terms are less frequent.

The methods EM, SimpleKMeans, XMeans, FarhestFirst and sIB were all applied to the datasets containing the answers to question 19 and question 6. EM and XMeans were run with no number of clusters specified. Furthermore, all methods were executed with the number of clusters to be created set to $2 \leq k \leq 6$. We have used EuclideanDistance as a distance function when needed. The random seed was set to 27 and 42 when this parameter was needed.

Because of the many dimensions in our data, presenting

[6] ftp://ftp.cs.cornell.edu/pub/smart/english.stop

[7] WEKA is a popular suite of machine learning software, see http://www.cs.waikato.ac.nz/ml/weka/

them in a comprehensible way is quite challenging. WEKA provides a visualization tool, which is a scatter plot containing all the instances. Even though this tool works intuitively and is capable of comparing any two dimensions, it does not give insight in the coherency of all the dimensions. Instead, we export the data to excel and use sorting and conditional formatting to visualize results. We use two methods for visualization of the clustering, one is instance based (attributes along the columns and the instances along the rows) and the other cluster based (clusters along the rows). An example of the latter can be seen in figure 2.

Analysis

Cluster evaluation metrics can be extrinsic, based on comparisons between the output of the clustering system and a *gold standard*. Since we do not have a gold standard (yet), we need to resort to intrinsic metrics. These are based on how close elements from one cluster are to each other, and how distant from elements in other clusters (Amigó et al., 2009). Furthermore, we have performed a meta-clustering to compare the clusterings of different algorithms and/or different runs of the same algorithm.

Many internal validation measures exist. We have chosen the 'index I' measure as described by (Maulik and Bandyopadhyay, 2002), which has a reasonable performance and is quite intuitive.

It is defined as:

$$I = (\frac{1}{NC} \times \frac{\sum_{x \in D} d(x,c)}{\sum_i \sum_{x \in C_i} d(x,c_i)} \times max_{i,j} d(c_i,c_j))^P$$

where D: data set; c: center of D; P: number of attributes (dimensionality) in D; NC: number of clusters; C_i: the i-th cluster; c_i: center of C_i; $d(x,y)$: (Euclidean) distance between x and y

A high I index corresponds to a good clustering. We computed this metric from 30 clusterings on the dataset 'question29': three methods $\{EM, KMeans, sIB\}$, five cluster sizes $\{2,3,4,5,6\}$, and two random seeds $\{27,42\}$. The respective values are plotted in figure 3.

Looking at figure 3, we can clearly see a correspondence between clustering quality and the number of clusters. Extrapolation of the negative correlation might even indicate that no natural partitioning exists in the data. Furthermore we see that the sIB algorithm tends to score worse than the other two. Besides, in some cases the random seed has quite some influence on the scores.

The I index provides means to compare different clusterings on the same dataset. We can use it to decide which clustering best matches the natural partitioning in the data. We can also use this technique for determining the proper number of clusters to aim for. But beside this, it doesn't tell us much about the nature of the data itself. The scores can be interpreted in relation to each other, but do not give an absolute measure.

On a higher level, we can compare the clusterings of different algorithms and/or different runs of the same algorithm. We are interested in deriving a consensus solution, presuming that if many clustering algorithms reveal the same structure, there must be some intrinsic partitioning in the data. This method is loosely based on the idea of Cluster Ensemble (Strehl and Ghosh, 2003). The technique we have used for this investigation is meta-clustering: we have run an EM clusterer with 13 clusterings (partitionings) as attributes (features). With the number of clusters unspecified, 9 clusters were created. We have also run the EM algorithm with the number of clusters set to 2 and 5. The resulting partitionings were unstable as well, which strengthens our belief that no partitioning can be found.

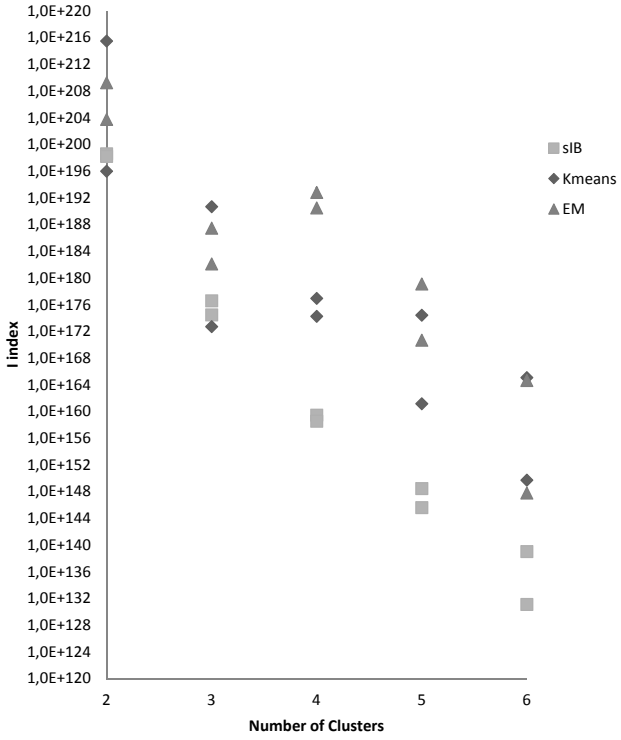

Figure 3: *I indices for 30 clusterings*

Cluster Tendency Although we did not find any indication of a natural grouping, the absence of it is hard to prove as we might have used the wrong technique or applied the wrong settings. The I index defines the quality of a clustering. Our objective is not to reveal the best possible clustering in the data however, but to investigate whether any clustering exist. "All clustering algorithms will, when presented with data, produce clusters - regardless of whether the data contain clusters or not. The first facet of a clustering procedure is actually an assessment of the data domain rather than the clustering algorithm itself. This is the field of *cluster tendency*, unfortunately this research area is relatively inactive" (Jain et al., 1999).

One method for assessing the cluster tendency of a set of objects is called VAT (Visual Assessment of (cluster) Tendency) (Bezdek and Hathaway,). First a distance matrix is created with the instances along both the axes, thus providing a pairwise (two-dimensional) interpretation of high-dimensional data. Secondly the instances are reordered according to an algorithm that is similar to Prim's algorithm for finding a *minimal spanning tree* of a weighed graph. Both matrices can then be displayed as *dissimilarity images*. The pairwise dissimilarity of the objects (the value in the distance matrix) determines the intensity or gray level of the corresponding pixel in the image. Clusters are indi-

Cluster:	size:	39-bag-imb	43-bag-retain	40-bag-incentiv	11-bag-negot	46-bag-tool	27-bag-schol	41-bag-leg	36-bag-reus	1-bag-ca	42-bag-part	38-bag-governm	44-bag-right	33-bag-level	26-bag-pol	13-bag-author	28-bag-common	84-bag-agr	88-bag-fund	45-bag-teach	2-bag-publ	24-bag-open	12-bag-research	57-bag-encour	58-bag-goal	79-bag-copyright	9-bag-purp
Cluster0	0,27	0	0	0	0,4	0	0	0,1	0,1	0,4	0,3	0	0,3	0	0	0,1	0,1	0,4	0,1	0,5	0,4	0	0,5	0	0,1	0,3	0,6
Cluster1	0,01		4,5	4,2	4,1	3,7	3,8	3,4	3,4	3,3	3,1	3,1	2,6	2,3	2,1	2,1	1,7	1,6	1,6	1,5	1,7	0	1,3	0	0	1,1	1
Cluster2	0,08	0	0	0,4	0	0,6	0,3	0	0,4	0,1	0	0,4	0	0,1	0,3	0,1	0,3	0,1	0,2	0,1	0,3	0,7	0,4	1	1	0,2	0,1
Cluster3	0,32	0	0	0	0,1	0	0	0,1	0	0,1	0	0	0,1	0,1	0	0,2	0	0	0	0,3	0,1	0	0,2	0	0	0,2	0,3
Cluster4	0,1	0	0	0	0,2	0	0,4	0	0	0	0,3	0,9	0,1	0	0,4	0,3	0	0	0,7	0	0,8	1,1	0,6	0,1	0	0,2	0,2
Cluster5	0,22	0	0,1	0	0	0	0,4	0,2	0,1	0	0	0	0,1	0,1	0,2	0,4	0,5	0,2	0,3	0,1	0,5	0,8	0,2	0	0,1	0,3	0
standard deviation		1,9	1,7	1,5	1,5	1,3	1,3	1,3	1,2	1,2	1,1	1,1	0,9	0,8	0,7	0,7	0,6	0,6	0,6	0,5	0,5	0,4	0,4	0,4	0,3	0,3	0,3

Figure 2: *Example of the proposed cluster based visualization in MS Excel*

cated by dark blocks of pixels along the diagonal. We have implemented this algorithm ourselves in R[8]. An example is displayed in figure 4. The distance measure we have used is Euclidean Distance. The intensity scale consisted of twelve shades of gray.

A dark cross appears in the top left corner of the ordered image. This corresponds to a part of the distance matrix containing zero values, which is of course the pairwise distance between two instances with zero values on all the features. A few of those instances exist, because of answers containing only function words (filtered out by the stop list) and very infrequent words (which are filtered out by the stringToWordVector filter in WEKA). Apart from these dark crosses, no dark blocks worth mentioning appear on the diagonal, which confirms that there is little or no cluster tendency in the data set.

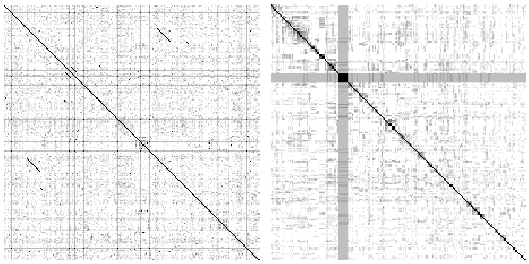

Figure 4: *Dissimilarity (left) and Ordered Dissimilarity (right) Image for Question 1*

6. Conclusions

We presented two experiments in attempting to detect arguments in replies to EU green papers. The first was aimed at classifying sentences as either argumentative or non-argumentative. From (Mochales and Moens, 2011) we learned that it should be feasible to automatically separate a text into argumentative and non-argumentative statements. Contrary to them we did not have a reasonably large tagged document set to train a machine learner. We resorted to a symbolic approach using keywords and regular expressions. Our classifier performs worse than theirs (F-score of 51 versus 73), probably partially due to difference in the type of documents. The Araucaria set that Mochales used is specifically aimed at argumentation and contains analysed arguments from newspapers, blogs and the like. Our set of replies to green papers is written in a far less argumentative style. Their second set consisted of documents extracted from legal texts of the European Court of Human Rights (ECHR), that has developed a standard type of reasoning and structure of argumentation over the years (Mochales and Moens, 2011). Our documents are written by different authors and their styles differ greatly.

In contrast to this first experiment, we found in our second series of experiments that semantic cohesion in the data is greater than cohesion based on linguistic constructs and vocabulary. This different result may have something to do with the different set of features used. Even though this result is promising, we must conclude that using content words in the answers to perform a clustering aiming at a semantic level of argument recognition was not feasible. This is partly due to the small size of the data set and the absence of a proper classification in the data. There appears to be no natural partitioning in the data, other than a very coarse topic-based division.

We are inclined to conclude that other features should be used to find any relevant grouping in this dataset. We will name a few possibilities here. Extending the work in our first experiment, the set of key words might be expanded with *argumentative phrases*, such as "First of all" or "as opposed to". Some research has been done on defining such phrases, see (van Eemeren et al., 2007) and (Knott and Dale, 1994). Some phrases may be grouped together, such as 'firstly' and 'secondly'. A related set of features could be created by tagging *sentiment phrases*, as has been described in (Fei et al., 2004).

One may also think of ways to tackle the problem of the small size of the data set. A model may be trained on an annotated argument corpus such as the Araucaria database. This would of course not take the specific terminology of a domain into account, but the model may be combined with a bag-of-words or an ontology to form a new model applying for both structural and symbolic classification. Furthermore, usage of the ART will lead to the creation of a corpus that can be used for future research.

To sum up, the results of our various experiments in automated support for finding and tagging arguments in natural language texts are not promising. The task seems too hard for the present state of the art, at least without a substantial corpus of tagged texts to use for training and testing.

The first step on this route therefore must be to set up such a corpus. The manual tagging of arguments using our tool is a logical step in that process. Making the ART available

[8] http://www.r-project.org/

as open source software, letting people tag arguments in responses to EU green papers and store these on our server will hopefully provide us with a usable corpus in the longer run. By making different levels of granularity available as described in section 2., the ART enables people to generate a gold standard at all these levels (that can be used as training and test set). This will enable experimentation with NLP techniques at any level. When automated support proves to be feasible, we can augment the existing user interface in such a way that users can benefit from it.

Acknowledgments
This research was partially funded by the EU in the Framework 7 programme (Grant Agreement No 247228) in the ICT for Governance and Policy Modeling theme (ICT-2009.7.3). Thanks to Sander Latour who helped with the first experiment described.

7. References

Amigó, E., Gonzalo, J., Artiles, J., and Verdejo, F. (2009). A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, pages 1–33.

Bezdek, J. and Hathaway, R.). VAT: a tool for visual assessment of (cluster) tendency. *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*, pages 2225–2230.

Brüninghaus, S. and Ashley, K. (2005). Generating Legal Arguments and Predictions from Case Texts. In *Proceedings of the Tenth International Conference on Artificial Intelligence and Law*, pages 65–74, New York, NY, USA. ACM Press.

Fei, Z., Liu, J., and Wu, G. (2004). Sentiment classification using phrase patterns. *The Fourth International Conference onComputer and Information Technology, 2004. CIT '04.*, pages 1147–1152.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, September.

Knott, A. and Dale, R. (1994). Using linguistic phenomena to motivate a set of coherence relations. *Discourse processes*, 18(1):35–62.

Marcu, D. and Echihabi, A. (2002). Unsupervised Approach to Recognizing Discourse Relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 368–375.

Maulik, U. and Bandyopadhyay, S. (2002). Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(12):1650–1654, December.

Mochales, R. and Moens, M.-F. (2011). Argumentation mining. *Artif. Intell. Law*, 19:1–22, March.

Moens, M.-F., Boiy, E., Mochales, R., and Reed, C. (2007). Automatic detection of arguments in legal texts. In *Proceedings of the 11th International Conference on Artificial Intelligence and Law*, pages 225–230, New York, NY, USA. ACM Press.

Moon, T. (1996). The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60.

Palau, R. M. and Moens, M.-F. (2009). Argumentation mining: the detection, classification and structure of arguments in text. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, ICAIL '09, pages 98–107, New York, NY, USA. ACM.

Slonim, N., Friedman, N., and Tishby, N. (2002). Unsupervised document classification using sequential information maximization. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '02*, page 129, New York, New York, USA. ACM Press.

Strehl, A. and Ghosh, J. (2003). Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions. *The Journal of Machine Learning Research*, 3:583–617.

van Eemeren, F. H., Houtlosser, P., and Snoeck Henkemans, a. F. (2007). Indicators of argument schemes. In Eemeren, F. H., Houtlosser, P., and Henkemans, A. F. S., editors, *Argumentative Indicators in Discourse*, volume 12 of *Argumentation Library*, chapter 6, pages 137–191. Springer Netherlands, Dordrecht.

Walton, D. (2002). In *Legal Argumentation and Evidence*. Pennsylvania State University Press.