Experiments in automated support for argument reconstruction
Winkels, R.G.F.; Douw, J.V.; Veldhoen, S.

Published in:
Fourteenth International Conference on Artificial Intelligence and Law: Rome, June 10-14, 2013, Consiglio Nazionale delle Richerche: proceedings of the conference

DOI:
10.1145/2514601.2514633

Citation for published version (APA):
This paper describes the outcomes of experiments in automated support for argument reconstruction from natural language texts. We investigated several possibilities to support a manual process by using natural language processing, from classifying pieces of text as either argumentative or non-argumentative to clustering text fragments in the hope that these clusters would contain similar arguments. Results are diverse, but also show that we cannot come far without an extensive pre-tagged corpus.

**Keywords**

argument mining, clustering, policy modelling

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 asks questions like “Should there be encouragement or guidelines for contractual arrangements between right holders and users for the implementation of copyright exceptions?”

Exploring and indexing these replies and their arguments from external sources is difficult and time consuming. EU FP7 project IMPACT’s goal is to provide means to support this process. 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 is to help the user by finding text fragments that contain arguments and possibly suggesting argument schemes that are used.

*Corresponding author.

1 From “Copyright in the Knowledge Economy”.


This paper focuses on two experiments in automated argument finding and reconstructing.

2. EXTRACTION OF ARGUMENTS

Manual extraction of arguments from a text is a non-trivial task. In [8], an example is given of three annotators that had to identify arguments in verdicts of the ECHR. 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.

2.1 Related Research

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. [9]). An actual attempt has been made by [8]. 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) 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 best done by machine learning classifiers.

A somewhat different approach is to start with classification of the relation between two text fragments rather than the text fragments themselves. [8] focus on the automated recognition of discourse relations, which are descriptions of how two spans of texts relate to each other. They used Naïve Bayes classifiers to distinguish between two relations, which had a performance of between 64% and 93%, depending on the relations that were compared.

3. 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

3 The European Court of Human Rights in Strasbourg, France.
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". 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 it is an argument at all. Domain knowledge and common sense is required to reconstruct the argumentation in these responses. Finally, almost every argument is an implicit “argument from position to know” [14]. 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.

3.1 Keywords and Regular Expressions

The first step was to see if the documents contained any keywords that indicate the use of argumentation. 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 arguments but do indicate the expression of an opinion which can indirectly indicate that an argumentation is used (e.g., essential, believe).

<table>
<thead>
<tr>
<th>Argumentation segments</th>
<th>BoF</th>
<th>Google</th>
<th>Ericsson</th>
<th>BEurope</th>
<th>ANRPI</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>however</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>7</td>
<td>16</td>
</tr>
<tr>
<td>thus / therefore</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td>lead(s) to / has resulted</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>in / result</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>conclude(s) / conclusion</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>assumption/assume</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>pointed out</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>at odds</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>since</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

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. When applying the first regular expression on our test set of 2 different documents the following confusion matrix was achieved:

<table>
<thead>
<tr>
<th>Tagging</th>
<th>Manual</th>
<th>Arg</th>
<th>Ntrtl</th>
<th>Ttl</th>
<th>Prec</th>
<th>Rec</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arg</td>
<td></td>
<td>16</td>
<td>7</td>
<td>23</td>
<td>69.6%</td>
<td>40.0%</td>
<td>50.8%</td>
</tr>
<tr>
<td>Neutral</td>
<td></td>
<td>24</td>
<td>65</td>
<td>89</td>
<td>73.0%</td>
<td>79.5%</td>
<td>76.1%</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>40</td>
<td>72</td>
<td>112</td>
<td>72.3%</td>
<td>72.3%</td>
<td>72.3%</td>
</tr>
</tbody>
</table>

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 (false positives). An obvious reason 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 premises (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.

4. 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, we created a pipeline to annotate the questions and answers in the documents after exporting them to plain text.

4.1 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 [4]. 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 \cite{10}. Another method is SimpleKMeans. It starts with a random initial partition and keeps assigning the patterns to clusters based on the similarity between the pattern and the cluster centers \cite{4}. 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 \cite{11}.

### 4.2 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 question 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.

The data was then loaded into WEKA Explorer\footnote{ftp://ftp.cs.cornell.edu/pub/smart/english.stop} 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 output-WordCounts 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. Three matching matrices were built relating the clusters to questions, sources, and topics. The latter is shown here for illustration:

<table>
<thead>
<tr>
<th>Cluster →</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topic ↓</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
</tr>
<tr>
<td>General</td>
<td>49</td>
<td>34</td>
<td>9</td>
<td>12</td>
<td>63</td>
</tr>
<tr>
<td>ELA</td>
<td>24</td>
<td>40</td>
<td>3</td>
<td>114</td>
<td>9</td>
</tr>
<tr>
<td>EPD</td>
<td>12</td>
<td>141</td>
<td>3</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>TR</td>
<td>17</td>
<td>36</td>
<td>87</td>
<td>2</td>
<td>38</td>
</tr>
<tr>
<td>UCC</td>
<td>61</td>
<td>15</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

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 \cite{4} 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:

<table>
<thead>
<tr>
<th>Classification</th>
<th>Precision</th>
<th>Recall</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question</td>
<td>0.123</td>
<td>0.309</td>
<td>0.176</td>
</tr>
<tr>
<td>Topic</td>
<td>0.420</td>
<td>0.219</td>
<td>0.288</td>
</tr>
<tr>
<td>source</td>
<td>0.027</td>
<td>0.232</td>
<td>0.049</td>
</tr>
</tbody>
</table>

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.

### 4.3 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, FarthestFirst 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 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 \cite{1}.

**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 \cite{4}. 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 \cite{7}, which has a reasonable performance and is quite intuitive. 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 \cite{2}.

Looking at figure \cite{2} 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 differ-
ent 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
[12]. The technique we have used for this investigation is
meta-clustering: we have run an EM clusterer with 13 clus-
terings (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.

Cluster Tendency.
Although we did not find any indication of a natural group-
ing, 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 clus-
tering 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, unfor-
unately this research area is relatively inactive” [4].

One method for assessing the cluster tendency of a set of
objects is called VAT (Visual Assessment of (cluster) Ten-
dency) [2]. 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 indicated by dark blocks of pixels along
the diagonal. We have implemented this algorithm ourselves
in R. An example ordered dissimilarity image is displayed in
figure 3. 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 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 infre-
quent words (which are filtered out by the stringToWord
Vector 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 3: Ordered Dissimilarity Image for Question 1

5. 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 [3] 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 [8]. 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 [13] and [5]. 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 [3].

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 symbolical 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 to hard for the present state of the art, at least without a substantial corpus of tagged texts to use for training and testing.

Acknowledgments

This research is 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.

6. REFERENCES