Using Clustering Techniques to Identify Arguments in Legal Documents

Prakash Poudyal
Department of Computer Science & Engineering
Kathmandu University
Dhulikhel, Nepal
prakash@ku.edu.np

Teresa Gonçalves
Department of Informatics
University of Evora
Evora, Portugal
tcg@uevora.pt

Paulo Quaresma
Department of Informatics
University of Evora
Evora, Portugal
pq@uevora.pt

ABSTRACT

A proposal to automatically identify arguments in legal documents is presented. In this approach, cluster algorithms are applied to argumentative sentences in order to identify arguments. One potential problem with this process is that an argumentative sentence belonging to one specific argument can also simultaneously be part of another, distinct argument. To address this issue, a Fuzzy c-means (FCM) clustering algorithm was used and the proposed approach was evaluated with a set of case-law decisions from the European Court of Human Rights (ECHR). An extensive evaluation of the most relevant and discriminant features to this task was performed and the obtained results are presented.

In the context of this work two additional algorithms were developed: 1) the "Distribution of Sentence to the Cluster Algorithm" (DSCA) was developed to transfer fuzzy membership values (between 0 and 1) generated by the FCM to a set of clusters; 2) the "Appropriate Cluster Identification Algorithm" (ACIA) to evaluate the proposed clusters against the gold-standard clusters defined by human experts.

The overall results are quite promising and may be the basis for further research work and extensions.

KEYWORDS

Machine Learning, Fuzzy clustering algorithm, argument mining, Legal documents, Natural Language Processing.

ACM Reference Format:
Prakash Poudyal, Teresa Gonçalves, and Paulo Quaresma. 2019. Using Clustering Techniques to Identify Arguments in Legal Documents. In *Proceedings of Third Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2019)*. Montreal, QC, Canada, 8 pages.

1 INTRODUCTION

Advances in communication technology, accessibility of media devices and mushrooming of social media has caused the number of individuals expressing opinions to grow exponentially. As a result, a massive amount of electronic documents is generated daily, including news editorials, discussion forums and judicial decisions containing legal arguments. In turn, rapid development of current research into argument mining is raising new challenges for natural language processing in various fields.

In general to automatically identify a legal argument within an unstructured text, three stages or modules are used in current practice. The first stage is to identify the argumentative and non-argumentative sentences, the second stage is to identify the boundaries of arguments and the third stage is to distinguish the argument's components (premise and conclusion).

To date, second stage processing has been performed by identifying the boundaries of arguments, an extensively explored method in the AI & Law literature. In this paper, we propose a clustering technique that groups argumentative sentences into a cluster of potential arguments with associated probabilities. An overview of our approach to the task is shown in Figure 1. The task is complex, since components of one argument (premise or conclusion) can also be involved in other arguments. In the example shown in figure 1 there are 2 distinct arguments (A and B). In this example sentence 2 belongs to argument A and also to Argument B. It is important to point out that, for instance, in the European Court of Human Rights corpus (ECHR) situations similar with this one (and even more complex) appear. Figure 2 shows a real example, where one sentence (marked in yellow) belongs to three arguments (7, 8, and 9) and is followed by another sentence, which belongs to a different argument (6), which is followed by another sentence belonging to arguments (7 and 8). To cluster such sentences, we propose to use a Fuzzy c-means (FCM) clustering algorithm [3] that provides a membership value ranging from 0 to 1 for each cluster of the sentence. These membership values are key assets of the FCM, since they allow us to associate each sentence with more than one cluster/argument. The performance of the FCM depends heavily on the selection of features that are used. In the context of our work we focused mainly on four kinds of features: N-gram, word2vec, sentence closeness and 'combined features'. Our aim is to identify the best performing set of features and techniques to cluster components to form an argument.

After extracting the features associated with each text, the FCM is used to obtain a cluster membership value for every sentence. To determine the composition of each cluster, we developed a specific algorithm: the "Distribution of Sentence to the Cluster Algorithm" (DSCA).

To evaluate the performance of the system, a second algorithm, the "Appropriate Cluster Identification Algorithm" (ACIA) was also developed to map each cluster of the system's output to the closest matching cluster in the gold-standard dataset.

The rest of the paper is organized as follows: section 3 contains a brief introduction to the datasets and to the measures used for evaluating the performance of the system. In Section 4 we describe

In: Proceedings of the Third Workshop on Automated Semantic Analysis of Information in Legal Text (ASAIL 2019), June 21, 2019, Montreal, QC, Canada.

Published at http://ceur-ws.org

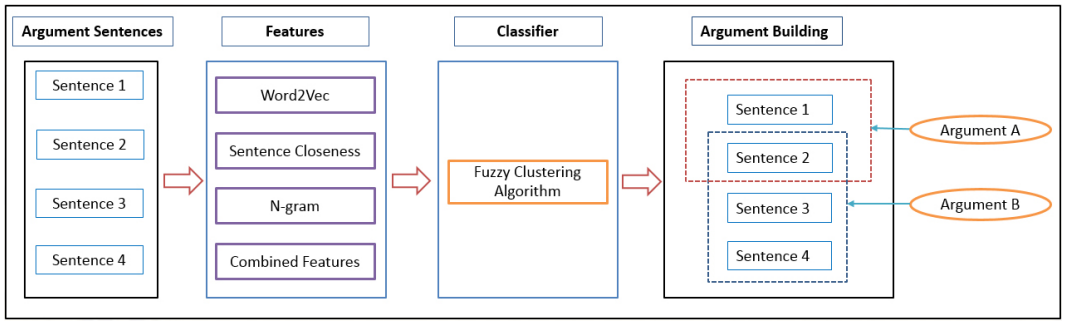

Figure 1: Overview of the Architecture

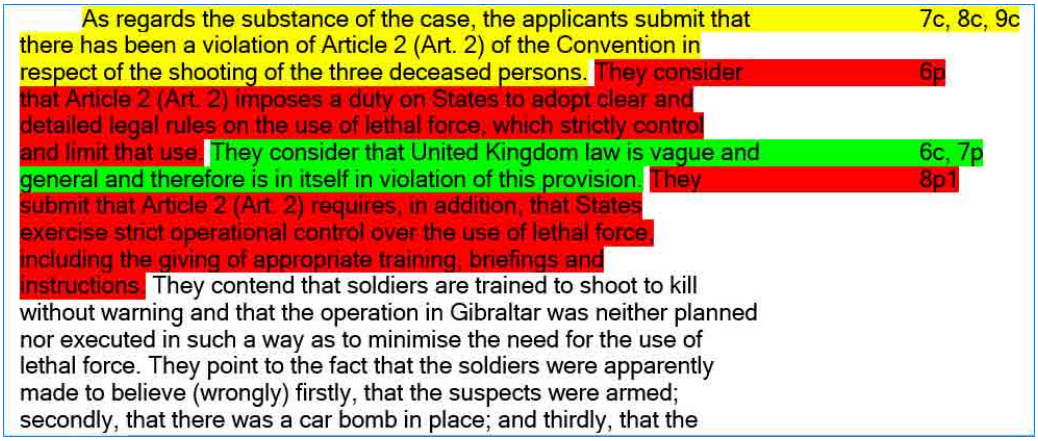

Figure 2: Example from the ECHR corpus

the proposed architecture including a description of features, a discussion on determining the optimum number of clusters, and the newly developed DSCA and ACIA algorithms. Section 5 evaluates the performance of all of our experiments. Lastly, Section 6 addresses conclusions and prospects for future work.

2 STATE OF THE ART

In the argument mining field, there has been very limited research about using clustering techniques to identify and group argumentative sentences into arguments. One of the most related research work was done by Rachel Mochales-Palau and M. Francine Moens [13, 14, 16]. They used the ECHR corpus, which was manually annotated and revised, and they were able to obtain around 80% accuracy in the detection of argumentative sentences using a statistical classifier. Then, they propose to detect argument limits using context-free grammars (CFG) to take into account the structure of documents or to use semantic distance measures to cluster related sentences. The CFG approach approach was applied to a limited subset of documents and obtained around 60% accuracy. However, they did not present any result for a semantic based approach. From the example in figure 2 it is clear that a CFG approach is not powerful enough to identify correctly the argument structure, as arguments can be interleaved and may not have a sequential structure. In this work we propose a clustering approach, which aims to overcome this restriction and is based in the relatedness of sentences.

In another related work, Sobhani et al. [24] have applied argument mining techniques to user comments aiming to perform stance classification and argument identification. Their work has quite different goals and they assume a predefined number of arguments, transforming the problem into a classification problem (tag sentences with the most adequate argument). They were able to obtain an f-measure of 0.49 for the argument tagging procedure. Moreover, user comments are typically simple sentences and do not have an inner argumentative structure.

J. Savelka and K. Ashley [22] have proposed to use machine learning techniques to predict the usefulness of sentences for the interpretation of the meaning of statutory terms. They explored the use of syntactical, semantic and structural features in the classification process and they were able to obtain an accuracy higher than 0.69.

Regarding argument relations, Stab and Gurevych [25] proposed an annotation scheme to model arguments and their relations. Their approach was to identify the relation (i.e. 'support' or 'attack') between the components of arguments. Their technique indicates which premises belong to a claim and constitute the structure of arguments.

Lawrence et al. [10] performed a manual analysis as well as an automated analysis to detect the boundaries of an argument. To train and test for automatic analysis, the authors relied on help from experts to analyze the text manually. For the automatic analysis, they used two Naive Bayes classifiers; one to identify the first word of a proposition and the other to identify the last word. [8], [21], and [17] continued this boundary approach, using the Conditional Random Fields (CRF) algorithm to segment the argument's components.

Lippi and Torroni [11] have a survey paper about the use of machine learning technologies for argument mining. The paper analyses several approaches made by different authors regarding argument boundary detection. This review article emphasizes that the boundary detection problems depend upon the adopted argument models. However, and as already referred, the non-sequential structure of arguments in the ECHR corpus creates new and complex problems, which can not be handled by simple boundary detection approaches.

Conrad [7] applied a k-means clustering algorithm over plaintiff claims in 'Premises Liability', 'Medical Malpractice' and 'Racial Discrimination' suits. The authors applied their technique to distinguish more effective plaintiff claims from less effective ones using an 'award_quotient' metric to segregate the claims. Besides award_quotient, the authors used features to help differentiate one cluster's properties from another. The authors mention that they also tried aggregative, partial and graphical features, but didn't find anything that yielded a performance superior to k-means.

3 CORPUS SELECTION AND EVALUATION PROCEDURES

We selected case-law documents from the European Court of Human Rights (ECHR)[1] annotated by R. Mochales [14]. The corpus is composed of 20 Decision and 22 Judgment categories released before 20 October 1999 by the European Commission on Human Rights. Both categories include similar information, however, the 'Decision' present the information briefly with an average word length of 3,500 words, whereas for 'Judgment' the average word length is 10,000 words. We have 9257 sentences, out of which 7097 (77%) of them are non-argumentative and 2160 (23%) argumentative sentences. Details about the ECHR corpus is available in [18].

Regarding evaluation, we used the standard Precision, Recall, and F-measure [2, 20] measures. Furthermore, we also used cluster purity to evaluate the quality of the obtained clusters. We computed the cluster purity [23] by counting the number of correctly assigned entities and dividing by the total number of N. Formally

$$ClusterPurity\,(\varphi, C) = \frac{1}{N}\sum_{d=1..k} max_{e=1..k}|w_d \cap c_e| \qquad (1)$$

where N is the summation of the total number of elements in all clusters, $\varphi = \{w_1, w_1, \cdots w_k\}$ is the set of clusters and $c = \{c_1, c_1, \cdots c_k\}$ is the set of classes. We interpret w_d as the set of sentences in w_d and c_e as the set of sentences in c_e in Equation 1.

[1] http://hudoc.echr.coe.int/sites/eng

4 SYSTEM ARCHITECTURE

Our proposal is to cluster argumentative sentences and, thereby, to identify legal arguments. As shown in figure 1 there are several phases: feature extraction; clustering algorithm; and argument building. In order to apply the fuzzy clustering algorithm we need first to identify the optimum number of clusters in the respective case-law file and, after running it, we need to convert the generate soft clustering values to hard clustering.

4.1 Feature Extraction

Typically features are values that represent a sentence and are suitable for a machine learning algorithm to handle. It is essential to select the most appropriate and precise features to train a machine learning algorithm so that the model can be successfully applied to new data. Therefore, good discriminant features are needed to correlate similarities between sentences and also address the sequential nature of sentences (since the majority of the components of an argument are presented in order). To address this requirement, the following features were used: N-gram [5], word2vec, and sentence closeness (discussed below). Another feature set can also be obtained by combining these three existing features into what we called "Combined Features". Each kind of features is discussed below.

Word2vec: The word2vec approach was proposed by [12] and can be implemented in two different ways: as a 'Continuous Bag of Words' (CBW) or as a 'Skip gram'. With Skip-grams, context words are predicted from selected words in the text, whereas with CBW, a word vector is predicted from the context of adjacent words. A Wikipedia dump of 05-02-2016 was used as input to the word2vec implementation of Gensim [19], where 100 dimension vectors were generated for each word. From the training set, each word of the sentence is looked up and its corresponding vector found among the generated word vectors. Then, the average of all vectors of the words presented in the sentence is taken and considered to be the 'sentence vector'.

Sentence Closeness: Sentence closeness is the reciprocal of the inter sentence distance (i.e. the distance between sentences) counted in units of whole sentences. To capture the sequential nature of sentences, distance is a useful feature that helps to determine which sentences belong to which argument. The highest scoring sentence is considered to be the origin sentence (with a score of 1) from which all other distances are measured. With the exception of the origin sentence, 'closeness' scores should decrease monotonically as they move away from the origin. Furthermore, as meaning and concepts flow from one sentence to another, this implies that sentences whose 'closeness' is high are good candidates for being clustered together i.e. they belong to the same argument. Equation 2 was used to calculate the 'closeness' for each pair of sentences.

$$Closeness(s_1, s_2) = \frac{1}{1 + |n(s_1) - n(s_2)|} \qquad (2)$$

where n is a function which calculates the number of sentences from the beginning of the text until the sentence of its argument.

Combined Features: The previously presented features (N-gram+'Sentence Closeness'+Word2vec) were combined into a new

feature in an attempt to improve the performance of the clustering algorithms.

4.2 Identification of the optimum number of clusters

An argument cluster is a set of sentences which together comprise a single, coherent legal argument. The process by which sentences are aggregated into arguments in this way is called clustering. To cluster sentences successfully into arguments, it is currently necessary to specify in advance how many clusters to expect within a corpus and until recently, there has been no well-established approach to defining this. Techniques that claim to be able to define the optimum number of clusters in the FCM have been proposed by [26] and Latent Dirichlet Allocation (LDA) [4].

Employing the Xie and Beni approach [26], we determined experimentally that an FCM with a fuzziness value of m set to 1.3 in concert with the features Word2Vec, N-gram and Sentence Closeness, yielded the best results with our particular set of case-law files. [26] technique selects the best candidate for the number of clusters after obtaining the minimum index value from that respective cluster number.

The Latent Dirichlet allocation (LDA) technique estimates the number of 'topics' existing within a text, which means estimating the probabilities of groupings within the text. Inspired by the concept, it was decided to look for such groups within our own corpora, and to use the estimated number of topics as a proxy for the number of clusters. We selected the 'CaoJuan2009' method as a metric which is the best LDA model based on density [15]. 'CaoJuan2009' was tested and agreed well with the number of topics (equivalent to our 'number of clusters') predicted by the minimum index value [6].

Figure 3 illustrates the results for each experiment: the first is the gold-standard, the second is using Xie and Beni's proposal, and the third is from LDA [6]. In the case of Xie and Beni, it can be observed that case-law files 02, 31, 32, 39, 42 find the closest number of clusters to the gold-standard, whereas for other case-law files the differences are greater.

In case of Cao *et al.*'s prediction: Case-law files 40 and 41 finds the correct data required for identification, whereas other case-law files present a slight difference, but not as big as that observed by Xie and Beni. The exact accuracy score achieved was an identical 8% for both LDA and Xie and Beni.

We also used equation 3, to calculate the difference between the number of clusters of the gold-standard (C_g) and the ones predicted by our system (C_s). If they differ only by one unit then we considered the prediction is "almost correct"; otherwise it's an incorrect prediction. The result of applying this filter shows that the accuracy of the closeness scores increase in value to 58% for LDA and 42% for Xie and Beni, respectively.

$$|C_s - C_g| \leq 1 \tag{3}$$

where C_s is the cluster number given by the prediction system, and C_g is the cluster number of the gold-standard.

From the analysis, it's possible to conclude that LDA achieves a greater accuracy than Xie and Beni and is much closer to the gold-standard. As a consequence, in our experiments we used this methodology to predict the adequate number of clusters. Improvements in the results are expected to be achieved in the future as more discriminant features are used.

4.3 Clustering Algorithm

After extracting the features, we used a standard Fuzzy c-means (FCM) Clustering algorithm [3] to generate membership values ranging from 0 to 1 for each cluster. The number of clusters was defined based on the algorithm proposed and described in section 4.2. We set the fuzziness value $m \in \{1.1, 1.3, 2.0\}$.

4.4 Distribution of Sentence to the Cluster Algorithm (DSCA)

The Distribution of Sentence to the Cluster Algorithm (DSCA) algorithm aims to transform the membership value generated by FCM (between 0 and 1) into a set of clusters (soft to hard clustering problem). The FCM output represents a membership probability indicating how likely it is that the sentence belongs to a particular cluster. DSCA is presented as Algorithm 1

Membership values are represented by a matrix where each row represents a sentence and each column is labelled with a cluster number (C_i) ranging from 1 to C. To assign a sentence to the respective cluster, a threshold value t needs to be specified to help define boundaries between the clusters. The cluster assignment is done only if the difference between the maximum membership value of the i^{th} position is less than the threshold value for the cluster, otherwise the sentence is rejected. The algorithm ends after conducting an iterative process through all positions in the matrix. The concept of *threshold value* is discussed by [1] as well as [9]. The authors claim that the definition of the appropriate *threshold value* should be determined by experimentation. After applying the DSCA algorithm, we were able to obtain a proposal for legal arguments: the identified clusters and their sentences.

5 RESULTS AND EVALUATION

In order to perform an evaluation of the performance of our system, we needed to find the best mapping between our system's clusters and the existing gold-standard data clusters from the ECHR. Therefore, we proposed and developed a new algorithm the "Appropriate Cluster Identification Algorithm "(ACIA) to solve this problem. This algorithm maps the argument predicted by our system to the closest matching argument in the gold-standard corpus. Here, we describe the details of the algorithm.

5.1 Appropriate Cluster Identification Algorithm (ACIA)

The ACIA algorithm aims to find the best mapping between the system's predicted clusters and the gold-standard dataset clusters. A formal description of the ACIA algorithm is presented in Appendix A but the general idea is the following:

- Select the best pair mapping between the clusters
- Remove these nodes from the set of clusters
- Iterate until there is no available pair of clusters
- The final mapping is composed by the set of the selected pair mappings

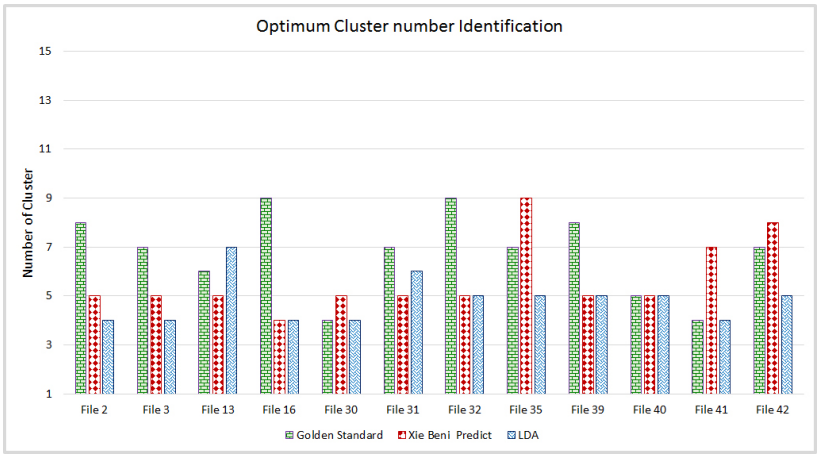

Figure 3: Argument numbers of gold-standard vs. System Prediction (proposed by Xie and Beni and Cao *et al.*)

Algorithm 1: Distribution of Sentence to the Cluster Algorithm (DSCA)

1. Denote the matrix of the sentences x cluster by $(a_{ij}) \in [0, 1]$, $i = 1, 2, 3 \cdots S$ and $j = 1, 2, 3 \cdots C$ such that i stands for sentence and j stands for cluster.
2. Pre-selected threshold (t) is defined
3. **for** *each i do* **do**
   $i_{max} = max(a_{ij}) \quad \forall i$
   for *each j do* **do**
      if $(i_{max} - a_{ij}) < t$ **then**
         select sentence i for cluster j
         else
            reject i;
         end
      end
   end

end

After identifying the best mapping, the f_1 measure is calculated for each cluster and the overall average f-measure value is obtained.

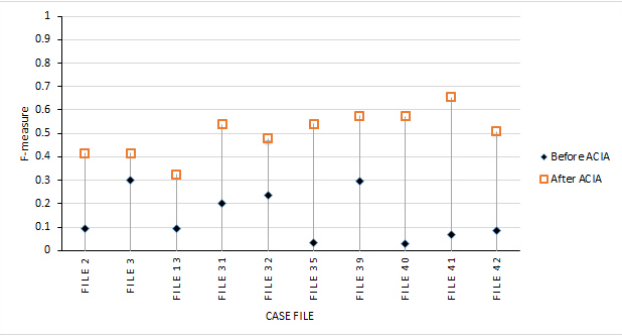

Figure 4: F_1 score before and after applying ACIA

In figure 4 we can see the relevance of performing an optimized mapping between the system's predicted clusters and the gold-data arguments. We can observe that the value of 'After ACIA' (square symbol) is higher (above 0.3 for all files), whereas in the case of a sequential mapping between the two clusters 'Before ACIA' (diamond symbol), the maximum value never exceeds is 0.3.

5.2 Performance Measurement

The experiment was conducted with the features mentioned in section 4.1 with fuzziness parameters $m \in \{1.1, 1.3, 2.0\}$ and *threshold value* $t \in \{0.0001, 0.00001, 0.000001\}$ used for the conversion from a soft to a hard clustering. For the reason of space, we present the results for the features and parameters that score the highest f_1 value in most of the case-law files. Table 1 presents the performance result (precision, recall and f_1, cluster purity) showing the N-gram, Sentence closeness, Word2vec and 'Combine features' using a *threshold value* $t = 0.00001$ and FCM fuzziness (m) = 1.3. Along with this, we include the number of sentences of each case-law file. The highest f_1 value of each case-law file obtained from each feature is highlighted in bold and underlined. Case-law files 03, 13, 16, 31, 32 and 42 obtained the highest value using Word2vec features. Case-law file 02 scored the highest f_1 value with N-gram, and case-law files 30, 35 and 41, the highest f_1 with the combined features. Likewise, case-law file 40 scored the highest f_1 value with the Sentence Closeness feature. From this analysis, we can conclude that Word2vec seems to be the best overall approach.

In comparison to Word2vec, N-gram did not perform as well. The main reason for this effect is that N-gram uses a bag of words approach which is not effective in finding similarities between sentences, and the results show that the performance of N-gram depends upon the number of sentences; if the number of the sentences in the case-law file is high, then N-gram performance is poor.

Sentence Closeness is another important feature that helps to understand the sequential context of the sentence. The sentence following an argumentative sentence often has a huge impact on the argument, as the meaning/context of a sentence usually flows sequentially. The results in this table show that the performance of

Case	#S	N-gram				Sentence Closeness				Word2vec				Combined Feature			
		Pre	Rec	f_1	Purity	Pre	Rec	f_1	Purity	Pre	Rec	f_1	Purity	Pre	Rec	f_1	Purity
02	15	0.698	0.485	**0.573**	**0.625**	0.342	0.221	0.268	0.412	0.656	0.367	0.470	0.563	0.625	0.450	0.523	0.600
03	15	0.619	0.429	0.506	0.563	0.405	0.333	0.366	0.412	0.714	0.429	**0.536**	**0.600**	0.524	0.381	0.441	0.533
13	20	0.508	0.628	0.561	0.500	0.413	0.344	0.375	0.400	0.602	0.581	**0.591**	**0.550**	0.342	0.344	0.343	0.400
16	33	0.125	1.000	0.222	0.125	0.437	0.481	0.458	0.424	0.449	0.449	**0.449**	**0.424**	0.125	1.000	0.222	0.125
30	25	0.265	1.000	0.419	0.263	0.252	0.275	0.263	0.320	0.351	0.363	0.357	**0.360**	0.272	1.000	**0.428**	0.275
31	15	0.317	0.714	0.439	0.313	0.524	0.571	0.547	0.533	0.595	0.524	**0.557**	**0.533**	0.429	0.500	0.462	0.400
32	17	0.335	0.785	0.470	0.326	0.481	0.393	0.433	0.474	0.648	0.485	**0.555**	**0.529**	0.500	0.396	0.442	0.474
35	13	0.429	0.414	0.421	0.467	0.619	0.414	0.496	0.571	0.667	0.414	0.511	0.615	0.845	0.636	**0.726**	**0.769**
39	17	0.352	0.588	**0.440**	0.346	0.400	0.431	0.415	**0.421**	0.362	0.525	0.429	0.368	0.310	0.613	0.412	0.250
40	14	0.400	0.370	0.384	0.467	0.587	0.530	**0.557**	0.533	0.519	0.520	0.520	**0.533**	0.400	0.420	0.410	0.467
41	12	0.517	0.563	0.539	0.500	0.625	0.625	0.625	**0.583**	0.438	0.438	0.438	0.417	0.683	0.625	**0.653**	**0.583**
42	18	0.464	0.440	0.452	0.389	0.433	0.414	0.424	0.389	0.643	0.486	**0.553**	**0.500**	0.431	0.598	0.501	0.414

Table 1: Precision, Recall and f_1, Cluster Purity value according to Case-law and the number of sentences

Sentence Closeness is satisfactory, but still lacking in comparison to Word2vec. The Combined feature also has an impact, as it is a combination of Word2vec, N-gram and Sentence Closeness. The Combined feature offered the highest f_1 value for those case-law files for which Word2vec did not offer significant results, with the exception of case-law files 02, 39 and 40. Overall, 66% of the case-law files obtained the highest f_1 using Word2vec and Combined feature.

Furthermore, in the case of N-gram and the Combined feature, we found recall is further elevated by up to 1, but precision is very low for case-law files that have a large number of sentences. This is because the n-gram feature is inappropriate for such case-law files. If the feature is not sufficiently discriminating enough to distinguish among sentence categories, applying the FCM provides equal membership probability values (or very close to equal) for every category, essentially providing no useful information. As a result, during the process of forming hard clusters, such sentences get equally distributed over all clusters.

Table 1 also presents the cluster purity value of each feature obtained for each case-law file. Word2vec was found to play the leading role in case-law files 03, 13, 16, 30, 31, 32, 40, and 42. Sentence Closeness scored highest in four case-law files: 16, 31, 39 and 41. However, case-law 16 and 31 tied with Word2vec. Overall the purity values are satisfactory, except in case-law file 16 and 33 with the Combined feature and N-gram. Case-law 16 which had 33 sentences had the lowest value (0.125) from Combined and N-gram features. Similarly, case-law 30, which has 25 sentences, obtained 0.275. On the other hand, case-law 35, which had 13 sentences, scored 0.726 (the highest value) using the Combined feature. From this analysis, we can conclude that having a greater number of sentences also affects the clustering quality negatively and that Word2vec is the dominant feature for obtaining acceptable f_1 and cluster purity values. It is apparent from the data in Table 1 that f_1 and cluster purity are well correlated.

Overall, the results obtained – average accuracy of 0.59, macro f-measure of 0.497 and cluster purity of 0.499 – from the proposed framework are quite promising, even if they cannot be easily compared with other researchers' results. The most related work is the one by Mochales and Moens [13, 16]; they obtained a 60% accuracy result in the argumentation structure detection task. It is important to refer they did not present the precision and recall measures and that they tried to handle a much more simple problem, because they assumed sequential argumentative structures.

Sobhani et al. [24] obtained a very similar f-measure value (0.49), but also with a much less complex task: classification of sentences from a predefined argument list.

Goudas [8] obtained an accuracy of 42%, while segmenting the argumentative sentences using Conditional Random Fields (CRF). Lawrence [10] precision and recall for identifying argument structure using automatically segmented propositions was 33.3% and 50.0%, respectively.

Stab and Gurevych [25] also encountered problems dealing with 'support' and 'attack' relations. The main reason for this was that their approach was unable to identify the correct target of a relation, especially in a paragraph with multiple claims or reasoning chains.

6 CONCLUSION AND FUTURE WORK

We proposed a new clustering technique for grouping argumentative sentences in legal documents. We also proposed and implemented an evaluation procedure for the proposed system and an approach to identify the total number of arguments in a case-law document. Overall, the results that we achieved are satisfactory and quite promising. The macro f_1 and average cluster purity score for system prediction using Word2vec feature in case-law files that have 4 to 8 arguments is 0.497 and 0.499 respectively.

For future work, we intend to add and evaluate more features, such as 'semantic similarity' ones, aiming to improve these results. Moreover, as an extension of this work we are working on: a) the identification, in each cluster/argument, of sentences acting either as a premises or conclusions; b) the creation of a graph representation of the argument structure of each document (attack, support, and rebuttal arguments).

ACKNOWLEDGEMENT

The authors would like to express deep gratitude to EMMA-WEST Project in the framework of the EU Erasmus Mundus Action 2 and Agatha Project SI IDT number 18022 (Intelligent analysis system of open of sources information for surveillance/crime control), ALENTEJO 2020 for their invaluable support. Further, the authors would also like to extend sincere thanks to the reviewers for their constructive comments and suggestions.

REFERENCES

[1] Moh'd Belal Al-Zoubi, Amjad Hudaib, and Bashar Al-Shboul. 2007. A fast fuzzy clustering algorithm. In *Proceedings of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases*, Vol. 3. 28–32.
[2] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. 1999. *Modern information retrieval*. Vol. 463. ACM press New York.
[3] James C Bezdek, Robert Ehrlich, and William Full. 1984. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences* 10, 2-3 (1984), 191–203.
[4] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.
[5] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics* 18, 4 (1992), 467–479.
[6] Juan Cao, Tian Xia, Jintao Li, Yongdong Zhang, and Sheng Tang. 2009. A density-based method for adaptive LDA model selection. *Neurocomputing* 72, 7-9 (2009), 1775–1781.
[7] Jack G. Conrad and Khalid Al-Kofahi. 2017. Scenario Analytics: Analyzing Jury Verdicts to Evaluate Legal Case Outcomes. In *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law (ICAIL '17)*. ACM, New York, NY, USA, 29–37. https://doi.org/10.1145/3086512.3086516
[8] Theodosis Goudas, Christos Louizos, Georgios Petasis, and Vangelis Karkaletsis. 2014. Argument extraction from news, blogs, and social media.. In *Hellenic Conference on Artificial Intelligence*. Springer, 287–299.
[9] A. K. Jain, M. N. Murty, and P. J. Flynn. 1999. Data clustering: a review. *ACM computing surveys (CSUR)* 31, 3 (1999), 264–323.
[10] John Lawrence, Chris Reed, Colin Allen, Simon McAlister, and Andrew Ravenscroft. 2014. Mining Arguments From 19th Century Philosophical Texts Using Topic Based Modelling. In *Proceedings of the First Workshop on Argumentation Mining*. Association for Computational Linguistics, Baltimore, Maryland, 79–87. https://doi.org/10.3115/v1/W14-2111
[11] Marco Lippi and Paolo Torroni. 2016. Argumentation mining: State of the art and emerging trends. *ACM Transactions on Internet Technology (TOIT)* 16, 2 (2016), 10.
[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their Compositionality. In *Advances in neural information processing systems*. 3111–3119.
[13] Raquel Mochales and Marie-Francine Moens. 2011. Argumentation mining. *Artificial Intelligence and Law* 19, 1 (2011), 1–22.
[14] Raquel Mochales-Palau and Marie-Francine Moens. 2007. Study on sentence relations in the automatic detection of argumentation in legal cases. *FRONTIERS IN ARTIFICIAL INTELLIGENCE AND APPLICATIONS* 165 (2007), 89.
[15] Nidhi. [n. d.]. Number of Topics for LDA on poems from Elliston Poetry Archive. Available at http://www.rpubs.com/MNidhi/NumberoftopicsLDA (2017-03-31).
[16] Raquel Mochales Palau and Marie-Francine Moens. 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)*. ACM, New York, NY, USA, 98–107. https://doi.org/10.1145/1568234.1568246
[17] Joonsuk Park, Arzoo Katiyar, and Bishan Yang. 2015. Conditional random fields for identifying appropriate types of support for propositions in online user comments. In *Proceedings of the 2nd Workshop on Argumentation Mining*. 39–44.
[18] Prakash Poudyal, Teresa. Goncalves, and Paulo. Quaresma. 2016. Experiments on identification of argumentative sentences. In *10th International Conference on Software, Knowledge, Information Management Applications (SKIMA)*. 398–403. https://doi.org/10.1109/SKIMA.2016.7916254
[19] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, 45–50. http://is.muni.cz/publication/884893/en.
[20] Mendes M.E.S. Rodrigues and L Sacks. 2004. A scalable hierarchical fuzzy clustering algorithm for text mining. In *Proceedings of the 5th international conference on recent advances in soft computing*. 269–274.
[21] Christos Sardianos, Ioannis Manousos Katakis, Georgios Petasis, and Vangelis Karkaletsis. 2015. Argument Extraction from News.. In *ArgMining@ HLT-NAACL*. 56–66.
[22] Jaromir Savelka and Kevin D Ashley. 2016. Extracting case law sentences for argumentation about the meaning of statutory terms. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*. 50–59.
[23] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. 2008. *Introduction to information retrieval*. Vol. 39. Cambridge University Press.
[24] Parinaz Sobhani, Diana Inkpen, and Stan Matwin. 2015. From argumentation mining to stance classification. In *Proceedings of the 2nd Workshop on Argumentation Mining*. 67–77.
[25] Christian Stab and Iryna Gurevych. 2014. Annotating Argument Components and Relations in Persuasive Essays. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*. Dublin City University and Association for Computational Linguistics, 1501–1510.
[26] Xuanli Lisa Xie and Gerardo Beni. 1991. A Validity Measure for Fuzzy Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 13, 8 (Aug. 1991), 841–847. https://doi.org/10.1109/34.85677

A APPENDIX

A.1 Appropriate Cluster Identification Algorithm (ACIA)

Let A be the system's cluster set and the B the gold standard cluster set, respectively, having a cardinality of n: $A = \{a_1, \cdots, a_n\}$ and $B = \{b_1, \cdots, b_n\}$. We define the matrix $F = \{f_{ij}\}$ where $f_{ij} = a_i b_j$ with $a_i \in A$ and $b_j \in B$. Here, $F = \{f_{ij}\}$ is the f-measure value calculated by taking cluster i from A and cluster j from B.
We denote by $(F)_{ij}$ the matrix formed from the matrix F by removing the j^{th} column and i^{th} row

State 1 : Initialization

$$F^o = (f_{ij})_{n \times n}$$

$$R^o(-1, -1) = \emptyset$$

i.e. Nodes are connected with the cost value C=0 to form a tree structure.

State 2 : From $k = 0$ to n, iterate. At each k step, we have $F^{(k)}(i, j)$ and $R^{(k)}(i, j)$

Find all maximum elements of $F^{(k)}(i, j)$
Let $M_k = \left\{(i, j) | f_{ij}^{(k)} \textit{is themaximum element of } F^{(k)}(i, j)\right\}$
i.e. Maximum f-measure value is selected and place in tree structure;

State 3: For each element $(i, j) \in M_k$, update route

$$R^{(k+1)}(i, j) = R^{(k)}(i, j) \cup \{(i, j)\}$$

and matrix

$$F^{(k+1)}(i, j) = \left(F^{(k)}\right)_{ij}$$

Do it for all elements (i, j) of M_k

Stop when $k = n$.

i.e. Procedure repeat again for other remaining values;

State 4 : {For each route, calculate total cost}

$$TC_{R^{(k)}(i,j)} = \sum_{(i,j)\in R^{(k)}(i,j)} f_{ij}$$

i.e. The total cost of each route is calculated.

State 5: Select one of the maximum values

$$TC_{R_o(i,j)},$$

and its route

$$R_o(i,j) = \{(i_1, j_1), \cdots, (i_n, j_n)\}$$

i.e. The route with the maximum scores is selected.

After identifying the appropriate cluster (argument) with respect to the gold-standard; an f-measure is calculated between the i^{th} cluster of the system as recommended by the ACIA and the j^{th} cluster of the gold-standard. After that, the average f-measure value is calculated.