A Two-Step Method based on Embedding and Clustering to Identify Regularities in Legal Case Judgements

(Discussion Paper)

Graziella De Martino[1], Gianvito Pio[1,2] and Michelangelo Ceci[1,2,3]

[1]Department of Computer Science - University of Bari Aldo Moro, Via Orabona, 4, 70125, Bari (Italy)
[2]Big Data Laboratory, National Interuniversity Consortium for Informatics, Via Ariosto, 25, 00185, Rome (Italy)
[3]Department of Knowledge Technologies, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana (Slovenia)

Abstract

In an era characterized by fast technological signs of progress that introduce new scenarios every day, working in the law field may appear very difficult if not supported by the right tools. In this paper, we discuss a recently submitted work that proposes a novel method, called PRILJ, that identifies paragraph regularities in legal case judgments to support legal experts during the redaction of legal documents. Methodologically, PRILJ adopts a two-step approach that first groups documents into clusters, according to their semantic content, and then identifies regularities in the paragraphs for each cluster. Embedding-based methods are adopted to properly represent documents and paragraphs into a semantic numerical feature space, and an Approximated Nearest Neighbor Search method is adopted to efficiently retrieve the most similar paragraphs with respect to the paragraphs of a document under preparation. Our extensive experimental evaluation, performed on a real-world dataset, proves the effectiveness and the efficiency of the proposed method even if documents contain noisy data.

Keywords

Legal Information Retrieval, Embedding, Clustering, Approximate Nearest Neighbor Search

1. Introduction

The legal sector is generally characterized by a slow response to the new scenarios that appear every day in the modern society. In this context, Artificial Intelligence (AI) methods can support the design of advanced (also automated) solutions to improve the efficiency of the processes in this field. Among the attempts in this direction, we can mention the work presented in [1], where the authors applied AI techniques to measure the similarity among legal case documents, that can be useful to speed up the identification and analysis of judicial precedents. Another relevant example is the work in [2], where the authors consider the semi-automation of some legal tasks, such as the prediction of judicial decisions of the European Court of Human Rights.

SEBD 2021: The 29th Italian Symposium on Advanced Database Systems, September 5-9, 2021, Pizzo Calabro (VV), Italy

graziella.demartino@uniba.it (Graziella De Martino); gianvito.pio@uniba.it (Gianvito Pio); michelangelo.ceci@uniba.it (Michelangelo Ceci)
0000-0002-3492-6317 (Graziella De Martino); 0000-0003-2520-3616 (Gianvito Pio); 0000-0002-6690-7583 (Michelangelo Ceci)

Following this line of research, in this discussion paper, we describe a novel method, called PRILJ, that identifies paragraph regularities in legal case judgements, to support legal experts during the redaction of legal documents. Methodologically, PRILJ adopts a two-step approach that first groups documents into clusters, according to their semantic content, and then identifies regularities in the paragraphs for each cluster. Embedding-based methods are adopted to properly represent documents and paragraphs into a semantic numerical feature space, and an Approximated Nearest Neighbor Search method is adopted to efficiently retrieve the most similar paragraphs. Therefore, given a (possibly incomplete or under preparation) document, henceforth called *target document*, PRILJ supports the retrieval of similar paragraphs appearing in a set of reference documents related to previous transcribed legal case judgments.

Document clustering has received a lot of attention by the research community, but together with the design of advanced algorithms [3, 4, 5, 6], the most critical aspect is in the design of a proper representation of the objects/items at hand [7, 8], as well as of similarity measures. In the literature we can find several document similarity measures implemented through *a)* network-based approaches [9, 10], *b)* text-based methods [11, 1] or *c)* hybrid approaches [11].

In this context, PRILJ has the main advantage of properly combining embedding methods, to catch the semantics, with a two-step approach, that consists in learning a different representation for each group of documents, rather than one single model. This aspect allows us to capture peculiarities of paragraphs according to the specific topic represented by each cluster of documents.

Our extensive experimental evaluation, performed on a real-world dataset, proves the effectiveness and the efficiency of the proposed method. In particular, its ability of modeling different topics of legal documents, as well as of capturing the semantics of the textual content, appear very beneficial for the considered task, and make PRILJ very robust to the possible presence of noise in the data.

2. Method

Before describing PRILJ, in the following, we provide some useful definitions:

- **Training set D_T**: a collection of legal judgments, represented as textual documents, adopted to train our models;
- **Reference set D_R**: a collection of legal judgments, represented as textual documents, from which we are interested to identify paragraph regularities;
- **Target document d**: a legal judgment (possibly under preparation) about which we are interested to identify paragraph regularities from D_R.

The training set and the reference set may fully (or partially) overlap i.e., $D_T = D_R$ (or $D_T \cap D_R \neq \emptyset$), namely, the set of documents adopted to train our models may be the same as (or overlap with) the collection from which we want to identify paragraph regularities with respect to the target document. Note that PRILJ is fully unsupervised and the target document d is never contained in either the training set or in the reference set (i.e., $d \notin (D_T \cup D_R)$).

The three phases of PRILJ are detailed in the following subsections.

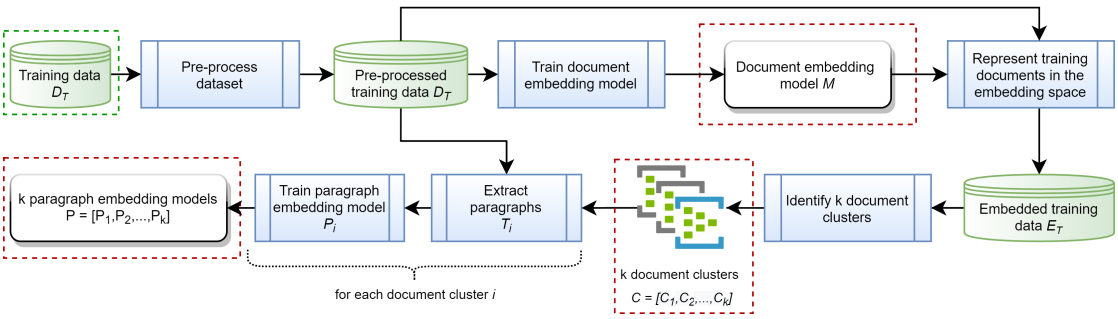

Figure 1: Graphical overview of the training phase. Green- and red-dotted rectangles represent inputs and outputs, respectively.

2.1. Training phase

As shown in Fig. 1, PRILJ starts with the application of some pre-processing steps to the documents in D_t. Specifically, the pre-processing consists of: *i)* lowercasing the text, *ii)* removing punctuation and digits, *iii)* applying lemmatization, and *iv)* removing rare words. The pre-processed documents are then used to train a document embedding model M, that is subsequently exploited to represent each document of the training set D_T in the latent feature space, obtaining the set of embedded training documents E_T. Such documents are then partitioned into k clusters $[C_1, C_2, ..., C_k]$ by adopting the k-means clustering algorithm. Each cluster of documents becomes the input for a further learning step at the paragraph level: documents falling in the same cluster will contribute to the learning of a specific paragraph embedding model. Algorithmically, for each document cluster $C_i, 1 \leq i \leq k$, we extract the paragraphs (i.e., sentences delimited by a full stop) from the documents falling into C_i and train a paragraph embedding model P_i. This approach allows us to learn more specific paragraph embedding models, according to the topic possibly represented by the identified clusters.

The embedding models, both at the document level and at the paragraph level, are learned by PRILJ through neural network architectures based on Word2Vec Continuous-Bag-of-Words (CBOW) [7] or Doc2Vec [8] distributed memory distributed memory (PV-DM). This choice is motivated by the fact that previous works demonstrated the superiority of Word2Vec and Doc2Vec over classical counting-based approaches, since they take into account both the syntax and semantics of the text [12, 1]. In addition, their ability to catch the semantics and the context of single words and paragraphs allow them to properly represent new (previously unseen) documents which features have not been explicitly observed during the training phase.

2.2. Paragraph embedding of the reference set

In Fig. 2, we show the workflow followed by PRILJ to represent the paragraphs of the documents belonging to the reference set into a latent feature space. Analogously to the training phase, we pre-process the documents of the reference set D_r. Then, each document of the reference set is embedded using the previously learned document embedding model M. The embedded representation of the document is then used to identify the closest document cluster that corresponds the optimal paragraph embedding model (i.e., P_c). We stress the fact that PRILJ

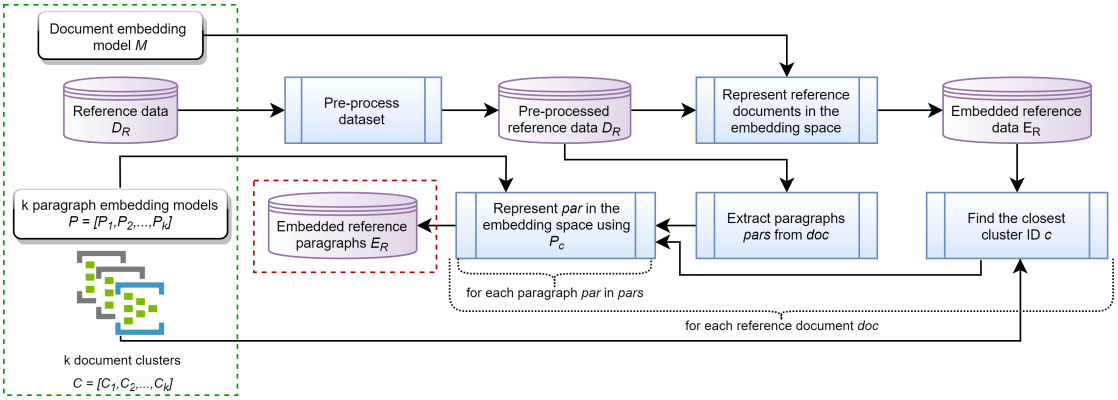

Figure 2: Graphical overview of the paragraph embedding of the reference set. Green- and red-dotted rectangles represent inputs and outputs, respectively.

performs this step to identify the most proper paragraph embedding model to represent the paragraphs of a given document.

The set of all the embedded paragraphs E_R is finally returned. Paragraph regularities for a given target document d will be identified from such set E_R.

2.3. Identification of paragraph regularities

The final phase, which workflow is represented in Fig. 3, starts by following the same steps mentioned in Sec. 2.2 to represent each paragraph of the target document d in the paragraph embedding space. Specifically, the most proper paragraph embedding model is adopted to embed its paragraphs, selected by identifying the closest document cluster with respect to d. For each embedded paragraph, we finally identify the top-n most similar paragraphs from the set of embedded paragraphs E_R belonging to the reference set.

It is noteworthy that their identification could straightforwardly be based on the computation of vector-based similarity/distance measures (e.g., cosine similarity, Euclidean distance, etc.) between the embedded paragraphs of the target document d and all the embedded paragraphs of the reference set E_r. Such a pairwise comparison would be computational intensive and would lead to inefficiencies during the adoption of the proposed system in a real-world scenario. To overcome this issue, we adopt a more advanced method for the identification of the top-n most similar paragraphs, based on random projections. In particular, we propose an approach based on Annoy [13], where the idea is to perform an approximated nearest neighbour search (ANNS), consisting in two phases: *index construction* on the paragraphs of the reference set, and *search*, that occurs when we actually need to identify the top-n most similar paragraphs with respect to a paragraph of the target document. During the index construction, we build T binary trees, where each tree is built by partitioning the input set of vectors recursively, by randomly selecting two vectors and defining a hyperplane that is equidistant from them. It is noteworthy that even if based on a random partitioning, vectors that are close to each other in the feature space are more likely to appear close to each other in the tree. During the search process, a priority queue is exploited, and each tree is recursively traversed, where the priority

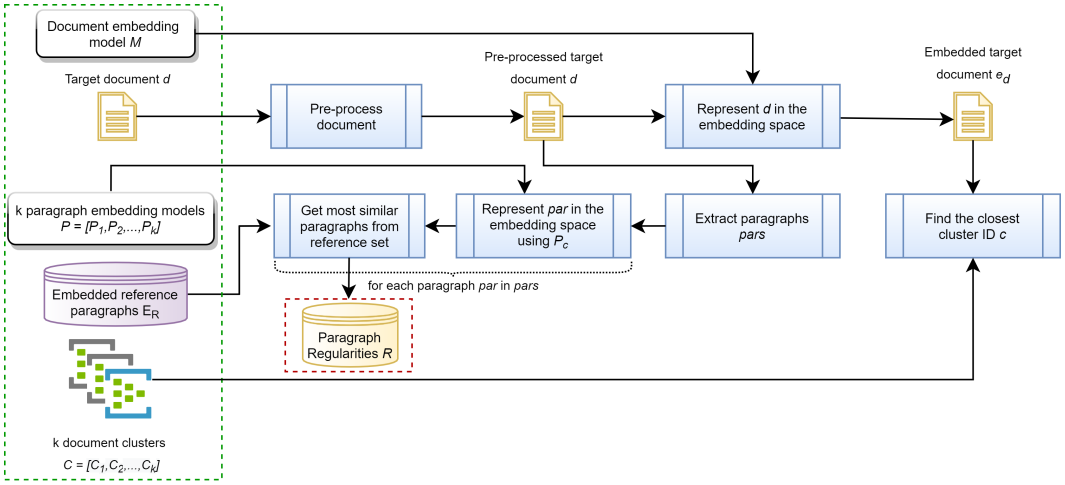

Figure 3: Graphical overview of the identification of paragraph regularities. Green- and red-dotted rectangles represent inputs and outputs, respectively.

of each split node is defined according to the distance to the query vector (that is a paragraph of the target document, in our case). This process leads to the identification of T leaf nodes, where the query vector falls into. The distance between the query vector and the set of vectors falling into the identified leaves is finally exploited to return the top-n most similar paragraphs [14].

3. Experiments

All the experiments were performed using a real-world dataset consisting of 4,181 official public EU legal documents, provided by EUR-Lex (https://eur-lex.europa.eu/homepage.html), in a 10-fold cross-validation setting. All the documents of the testing set were considered as target documents, while the reference set was built by constructing 20 replicas of each paragraph of the documents in the testing set, perturbed by introducing a controlled amount of noise. In particular, noise was introduced by replacing a given percentage of words of each paragraph by random words selected from the Oxford dictionary (raw.githubusercontent.com/cduica/Oxford-Dictionary-Json/master/dicts.json). In our experiments, we considered different levels of noise, namely, 10%, 20%, 30%, 40%, 50% and 60%, in order to evaluate the robustness of the proposed approach to different amounts of noise.

In order to assess the specific contribution of the adopted embedding strategies, we compared the results obtained through Word2Vec and Doc2Vec with those achieved using a baseline strategy, i.e., the classical TF-IDF. In all the cases, we adopted a 50-dimensional feature vector. Note that we use 50 features, since it is a commonly used dimensionality in other pre-trained embedding models. For TF-IDF, we selected the top-50 words showing the highest frequency across the set of legal judgments.

We specifically evaluated the contribution of the two-step model implemented in PRILJ with different numbers of clusters, i.e., $k \in \{\sqrt{|D_T|}/2, \sqrt{|D_T|}, \sqrt{|D_T|} \cdot 2\}$, and compared the observed performance with that obtained without grouping training documents into clusters (henceforth denoted as *one-step model*).

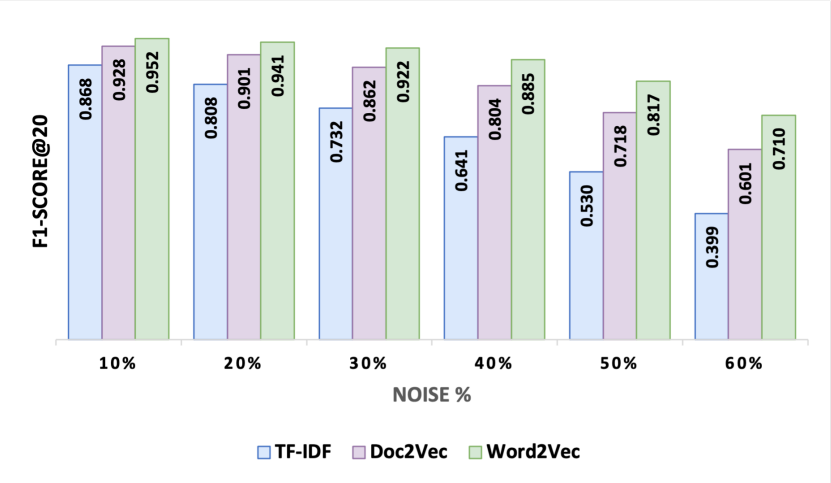

Figure 4: F1-score@20 results obtained when using TF-IDF, Doc2Vec or Word2Vec as embedding strategies, with the two-step model ($k = \sqrt{|D_T|} \cdot 2$).

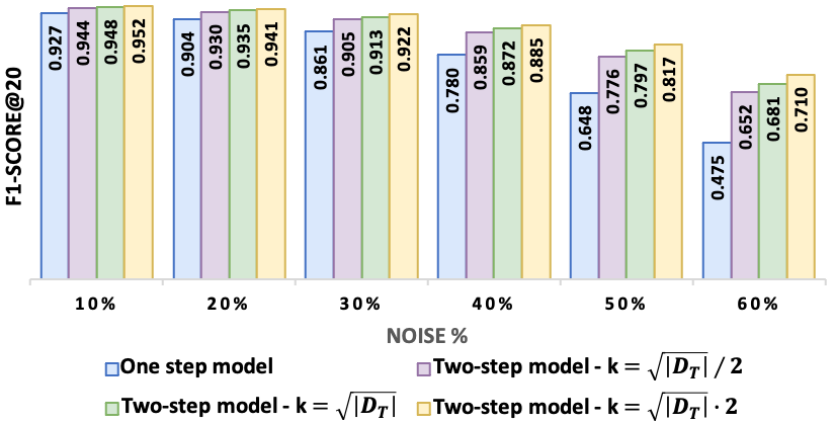

Figure 5: F1-score@20 results obtained with the two-step model (with different values of k) and with the one-step model. As embedding strategy, we considered Word2Vec.

Finally, we evaluated the effectiveness and the efficiency of the approach implemented in PRILJ for the identification of the *top*-n most similar paragraphs based on ANNS (with $T = 100$). Specifically, we performed an additional comparative analysis against a non-approximated solution based on the cosine similarity, on a subset of 100 documents randomly selected from the dataset. This analysis was performed considering the best number of clusters k, and also focused on evaluating the advantages in terms of computational efficiency.

As evaluation measures, we collected precision@n, recall@n and f1-score@n, averaged over the paragraphs of target documents and over the 10 folds, with $n \in \{5, 10, 15, 20, 50, 100\}$. Specifically, for each paragraph of a target document in the testing set, we considered as True Positives the number of correctly retrieved (perturbed) replicas from the reference set. Note that, in this discussion paper, for space constraints we only show the results in terms of f1-score@20.

	ANNS	Cosine Similarity
TF-IDF	**0.513**	407.612
Doc2Vec	**0.551**	580.842
Word2Vec	**0.610**	668.040

Table 1
Average running time (s) for the identification of the top-n most similar paragraphs, with the two-step model and $k = \sqrt{|D_T|} \cdot 2$.

3.1. Results

In Fig. 4 we can observe that, although the baseline based on TF-IDF obtained acceptable results, the adoption of the embedding methods implemented in PRILJ is significantly beneficial. Moreover, although Doc2Vec is natively able to work with word sequences, Word2Vec always obtains better results. This is possibly due to the fact that several paragraphs of different legal documents may share a similar topic, and the adoption of the unique sequence ID to associate the context with the document, as done by Doc2Vec (see [8] for details), may lead to overfitting issues.

In Fig. 5, it is possible to clearly observe the contribution of the two-step process we propose. Indeed, the results show that the proposed two-step model outperforms the one-step model, in all the situations. In particular, the two-step model is much more robust to the presence of noise: although we can still observe a decrease when the noise amount increases, its impact is much less evident. We can also observe that in general, the number of extracted cluster k seems to not significantly affect the results, even if the best results are observed with $k = \sqrt{|D_T|} \cdot 2$. This means that the documents are distributed among several topics and that learning a different (more specialized) paragraph embedding model for each of them is helpful to retrieve significant paragraph regularities.

Finally, the comparison between the adopted ANNS and the exact computation of the cosine similarity emphasized a difference of 0.6% in terms of f1-score@n, which can be considered negligible. On the other hand, the advantage in terms of efficiency is significant: the exact search required up to 1000x the time took by the ANNS implemented in PRILJ (see Table 1).

4. Conclusions

In this work, we discussed PRILJ, a novel approach to identify paragraph regularities in legal judgments. PRILJ represents documents and paragraphs thereof in a numerical feature space by exploiting embedding methods able to catch the context and the semantics. Moreover, PRILJ is based on a two-step model, that groups similar documents into clusters and, for each of them, learns a specific paragraph embedding model. This approach allows us to properly catch peculiarities exhibited by paragraphs and documents of similar topics and to handle the presence of noise in a robust manner. Finally, PRILJ is able to identify paragraph regularities very efficiently, thanks to an ANNS strategy.

Our extensive experimental evaluation has shown the accuracy and the efficiency of the developed approach on real data. This means that PRILJ can be considered a useful tool in real-world scenarios, also when large collections of legal documents have to be analyzed.

Acknowledgements

GP acknowledges the support of Ministry of Universities and Research through the project "Big Data Analytics", AIM 1852414-1 (line 1).

References

[1] A. Mandal, R. Chaki, S. Saha, K. Ghosh, A. Pal, S. Ghosh, Measuring similarity among legal court case documents, in: Proc. of the 10th Annual ACM India Compute Conference, Association for Computing Machinery, 2017, p. 1–9.

[2] M. Medvedeva, M. Vols, M. Wieling, Using machine learning to predict decisions of the european court of human rights, Artificial Intelligence and Law 28 (2020).

[3] P. Berkhin, Survey of clustering data mining techniques, A Survey of Clustering Data Mining Techniques. Grouping Multidimensional Data: Recent Advances in Clustering. 10 (2002).

[4] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining, KDD'96, 1996, p. 226–231.

[5] G. Pio, M. Ceci, C. Loglisci, D. D'Elia, D. Malerba, Hierarchical and Overlapping Co-Clustering of mRNA: miRNA Interactions, in: ECAI 2012, volume 242 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2012, pp. 654–659.

[6] R. Corizzo, G. Pio, M. Ceci, D. Malerba, DENCAST: distributed density-based clustering for multi-target regression, J. Big Data 6 (2019) 43.

[7] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, Advances in Neural Information Processing Systems 26 (2013).

[8] Q. Le, T. Mikolov, Distributed representations of sentences and documents, 31st International Conference on Machine Learning, ICML 2014 4 (2014).

[9] S. Kumar, P. K. Reddy, V. B. Reddy, A. Singh, Similarity analysis of legal judgments, in: Proceedings of the 4th Bangalore Annual Compute Conference, Compute 2011, Bangalore, India, March 25-26, 2011, ACM, 2011, p. 17.

[10] A. Minocha, N. Singh, A. Srivastava, Finding relevant indian judgments using dispersion of citation network, in: Proceedings of the 24th International Conference on World Wide Web, Association for Computing Machinery, 2015, p. 1085–1088.

[11] S. Kumar, P. K. Reddy, V. B. Reddy, M. Suri, Finding similar legal judgements under common law system, in: Databases in Networked Information Systems, Springer Berlin Heidelberg, 2013, pp. 103–116.

[12] K. Donghwa, D. Seo, S. Cho, P. Kang, Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec, Information Sciences 477 (2018).

[13] E. Bernhardsson, Annoy at github, https://github.com/spotify/annoy, 2015.

[14] W. Li, Y. Zhang, Y. Sun, W. Wang, W. Zhang, X. Lin, Approximate nearest neighbor search on high dimensional data - experiments, analyses, and improvement (v1.0), CoRR (2016).