Identification of Paragraph Regularities in Legal Judgements through Clustering and Textual Embedding

Graziella De Martino[1][0000−0002−3492−6317] and
Gianvito Pio[1,2][0000−0003−2520−3616]

[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)

Abstract. In an era characterized by fast technological progresses, working in the law field is very difficult if not supported by the right tools. In this paper, we present a novel method, called JPReg, that identifies paragraph regularities in legal case judgments to support legal experts during the preparation of new legal documents (i.e., paragraphs of existing documents that are similar to those of a document under preparation). JPReg adopts a two-step approach that first clusters similar documents, according to their semantic content, and then identifies regularities in the paragraphs for each cluster. Text embedding methods are adopted to represent documents and paragraphs into a numerical feature space, and an Approximated Nearest Neighbor Search method is adopted to efficiently retrieve the most similar paragraphs with respect to those of a target document. Our extensive experimental evaluation, performed on a real-world dataset, shows the effectiveness and the computational efficiency of the proposed method even in presence of noise in the data.

Keywords: Legal Information Retrieval · Embedding · Clustering · Approximate Nearest Neighbor Search

1 Introduction

The legal sector is generally characterized by a slow response to new scenarios that appear every day in the modern society. In this context, the adoption of Artificial Intelligence (AI) methods can be helpful to improve the efficiency of the processes in this field. Among the several attempts that we can find in the literature in this direction, we can mention the work presented in [13], 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 [14], where the authors considered the semi-automation of some legal tasks, such as the prediction of judicial decisions of the European Court of Human Rights.

Following this line of research, in this paper, we present a novel method, called JPReg (Judgement Paragraph REGularities), that identifies paragraph regularities in legal case judgements, to support legal experts during the preparation of new legal documents. Methodologically, JPReg is based on 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. Documents and paragraphs are represented into a semantic numerical feature space through text embedding methods, while the retrieval of similar paragraphs relies on an Approximated Nearest Neighbor Search (ANNS) approach, that provides a significantly higher computational efficiency with respect to classical similarity/distance-based methods. Therefore, given a (possibly incomplete or under preparation) document, henceforth called *target* document, JPReg 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 (e.g., for distributed computation, or co-clustering) [3, 7, 8, 18], the most critical aspect is in the design of a proper representation of the objects/items at hand [11, 16], as well as of similarity measures [10, 13, 17].

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

Our extensive experimental evaluation, performed on a real-world dataset, proves the effectiveness and the computational 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 JPReg very robust to the possible presence of noise in the data and capable to significantly outperform state-of-the-art competitors.

2 The proposed method JPReg

Before describing JPReg, 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

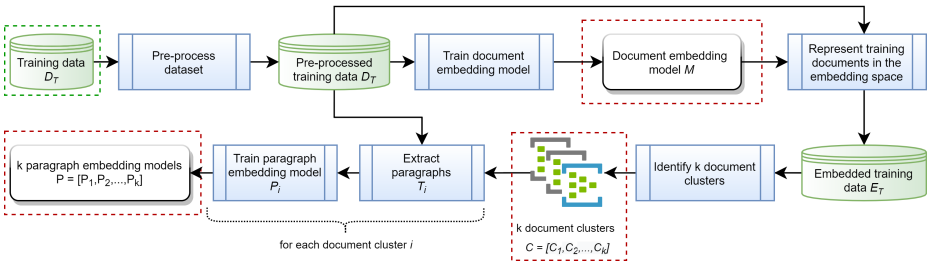

Fig. 1. Workflow followed by JPReg in the training phase. Green- and red-dotted rectangles represent inputs and outputs, respectively.

to identify paragraph regularities with respect to the target document. Note that JPReg is fully unsupervised and the target document d is never contained in either the training set or in the reference set (i.e., $d \notin (D_T \cup D_R)$).

JPReg consists of the three main phases, which are detailed in the following.

2.1 Training of document and paragraph embedding models

In Fig. 1, we show the workflow followed by JPReg in the first phase. JPReg first applies some pre-processing steps to the documents in D_t, namely: *i)* lowercasing of the text, *ii)* removal of the punctuation and digits, *iii)* lemmatization, and *iv)* removal of 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 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 JPReg through neural network architectures based on Word2Vec Continuous-Bag-of-Words (CBOW) [16] or Doc2Vec [11] distributed memory (PV-DM). Their adoption is motivated by the fact that previous works demonstrated the superiority of Word2Vec and Doc2Vec over classical counting-based approaches, such as TF-IDF, since they take into account both the syntax and the semantics of the text [6, 13]. In addition, their ability to capture 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.

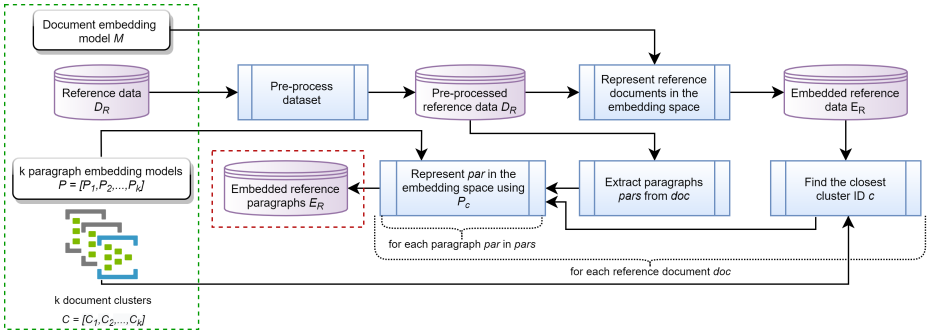

Fig. 2. Workflow followed by JPReg for the paragraph embedding of the reference set. Green- and red-dotted rectangles represent inputs and outputs, respectively.

2.2 Embedding of the paragraph of the reference set

In Fig. 2, we show the workflow followed by JPReg 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 is embedded using the previously learned document embedding model M. The embedded representation of the document is used to identify the closest document cluster that corresponds to the optimal paragraph embedding model (i.e., P_c), that can be adopted for its paragraphs. We stress the fact that this step performed by JPReg allows it to identify the most proper paragraph embedding model that was learned from a subset of the paragraphs possibly related to a similar topic.

The set of all the embedded paragraphs E_R is finally returned by this phase. Paragraph regularities for a given target document d under preparation 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.

As mentioned in Sec. 1, their identification could straightforwardly be based on the computation of vector-based similarity/distance measures (e.g., cosine similarity, Euclidean distance, etc.) between the identified numerical representation of the paragraphs of the target document d and that of all the embedded paragraphs of the reference set E_r. However, in a real-world scenario, such a pairwise comparison would be computational intensive and would lead to inefficiencies and delays. To overcome this issue, we adopt an approximated and highly

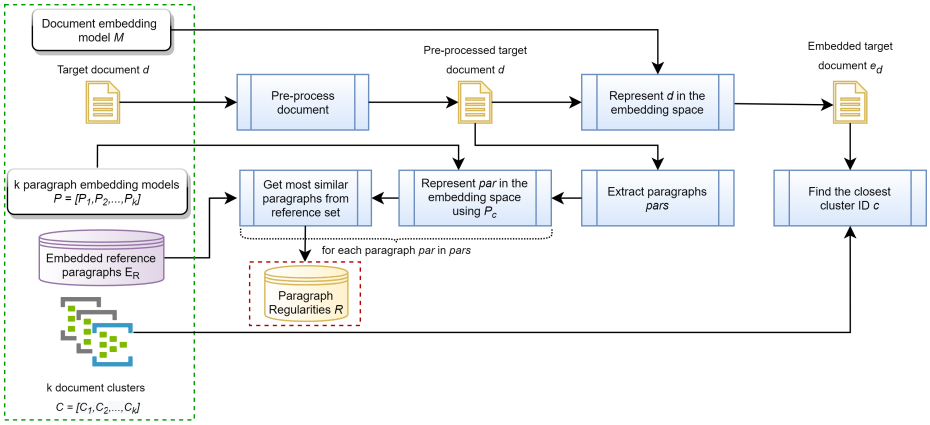

Fig. 3. Workflow followed by JPReg in the identification of paragraph regularities. Green- and red-dotted rectangles represent inputs and outputs, respectively.

efficient approach for the identification of the top-n most similar paragraphs, that is based on random projections. In particular, we propose an approach based on Annoy [1], where the idea is to perform an approximated nearest neighbor search (ANNS), which consists of 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 this process, a priority queue is exploited, and each tree is recursively traversed, where the priority of each split node is defined according to the distance to the query vector (here, a paragraph of the target document). This process leads to the identification of T leaves, where the query vector falls into. The distance between the query vector and the vectors falling into the such leaves is then exploited to return the top-n most similar paragraphs [12].

The adoption of this approach reduces the time complexity of the search phase for each paragraph of the target document from $O(|E_r|)$, in the case of the adoption of classical NNS approaches, to $O(log_2(|E_r|))$, that is the average length of a path in the trees from the root to a leaf node.

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[3], in a 10-fold

[3] https://eur-lex.europa.eu/homepage.html

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, as done in [4]. In particular, the noise was introduced by replacing a given percentage of words of each paragraph by random words selected from the Oxford dictionary[4]. In our experiments, we considered different levels of noise, namely, 10%, 20%, 30%, 40%, 50% and 60%, in order to assess the robustness of the proposed approach to different amounts of noise.

In order to quantify the specific contribution of the adopted embedding strategies, we compared the results obtained through Word2Vec and Doc2Vec with those achieved using a baseline approach, 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 evaluated the contribution of the JPReg two-step model with different numbers of clusters, i.e., with $k \in \{\sqrt{|D_T|}/2, \sqrt{|D_T|}, \sqrt{|D_T|} \cdot 2\}$, and compared the observed performance with that obtained without grouping training documents into clusters (henceforth denoted as *one-step model*).

We also performed an additional comparison with state-of-the-art methods. Specifically, we compared JPReg with **LEGAL-BERT-EURLEX**, that is the LEGAL-BERT model[5] fine-tuned by [2] using the EUR-LEX dataset, and **BERT-PLI**[6] which is based on BERT, fine-tuned with a small set of legal documents, proposed by [21] in the Competition On Legal Information Extraction/Entailment (COLIEE). Note that these competitors are embedding models, which are able to represent paragraphs as numerical feature vectors taking into account the semantics and the context of the textual content. Specifically, both LEGAL-BERT-EURLEX and BERT-PLI represent paragraphs in a 768-dimensional feature space. The embedding of each paragraph was computed as the mean of the embedding of its tokens.

Finally, we evaluated the effectiveness and the computational efficiency of the ANNS approach implemented in JPReg for the identification of the *top*-n most similar paragraphs. 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.

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 paper, for space constraints we only show the results in terms of F1-score@20.

[4] raw.githubusercontent.com/cduica/Oxford-Dictionary-Json/master/dicts.json
[5] https://huggingface.co/nlpaueb/legal-bert-base-uncased
[6] https://github.com/sophiaalthammer/bert-pli

3.1 Results

In Tab. 1, we report the F1-score@20 for the baseline one-step model and for the two-step model with different values of k, considering different embedding strategies and different levels of noise. From the table, we can observe that, although the TF-IDF led to acceptable results, the adoption of the embedding methods implemented in JPReg 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 [11] for details), may lead to overfitting issues.

From Tab. 1 and from Fig. 4, 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 lower F1-score when the noise amount increases, its impact is much less evident. We can also observe that in general, the number of clusters 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 *specialized* paragraph embedding models is helpful to retrieve significant paragraph regularities.

Focusing on the comparison with state-of-the-art systems, in Table 2 we report the F1-score@20 results obtained by JPReg (two-step model, $k = \sqrt{|D_T|} \cdot 2$, Word2Vec) and by the considered competitors, with different levels of noise.

One-step model

	Noise %					
	10%	**20%**	**30%**	**40%**	**50%**	**60%**
T	0.696	0.587	0.456	0.305	0.154	0.047
D	0.885	0.815	0.687	0.510	0.327	0.180
W	**0.927**	**0.904**	**0.861**	**0.780**	**0.648**	**0.475**

Two-step model - $k = \sqrt{|D_T|}/2$

	Noise %					
	10%	**20%**	**30%**	**40%**	**50%**	**60%**
T	0.835	0.764	0.678	0.573	0.444	0.291
D	0.918	0.889	0.847	0.779	0.675	0.533
W	**0.944**	**0.930**	**0.905**	**0.859**	**0.776**	**0.652**

Two-step model - $k = \sqrt{|D_T|}$

	Noise %					
	10%	**20%**	**30%**	**40%**	**50%**	**60%**
T	0.854	0.789	0.709	0.611	0.492	0.349
D	0.923	0.896	0.856	0.795	0.703	0.575
W	**0.948**	**0.935**	**0.913**	**0.872**	**0.797**	**0.681**

Two-step model - $k = \sqrt{|D_T|} \cdot 2$

	Noise %					
	10%	**20%**	**30%**	**40%**	**50%**	**60%**
T	0.868	0.808	0.732	0.641	0.530	0.399
D	0.928	0.901	0.862	0.804	0.718	0.601
W	**0.952**	**0.941**	**0.922**	**0.885**	**0.817**	**0.710**

Table 1. F1-score@20 results obtained with different embedding strategies (T=TF-IDF; D=Doc2Vec; W=Word2Vec) and different levels of noise. The upper-left subtable shows the results obtained with the one-step model, while the other subtables show the results obtained by JPReg with different numbers of clusters. The best result in a given subtable is shown in boldface, while the absolute best result is underlined.

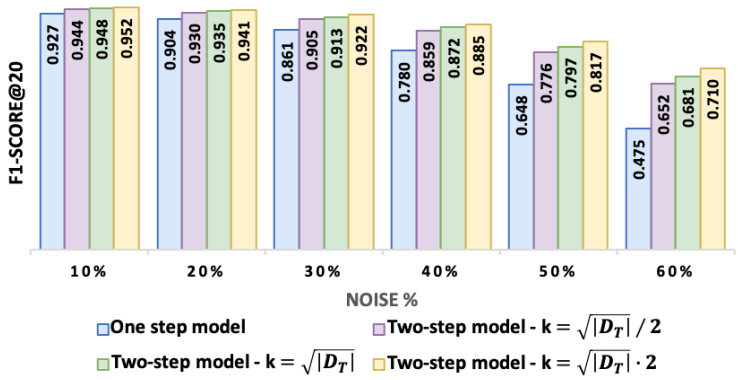

Fig. 4. F1-score@20 results obtained with the two-step model (with different values of k) and with the one-step model. As embedding strategy, we considered Word2Vec.

	Noise %					
	10%	**20%**	**30%**	**40%**	**50%**	**60%**
JPReg	**0.952**	**0.941**	**0.922**	**0.885**	**0.817**	**0.710**
LEGAL-BERT-EURLEX	0.820	0.446	0.172	0.063	0.025	0.010
BERT-PLI	0.432	0.078	0.024	0.011	0.007	0.004

Table 2. F1-score@20 results obtained by JPReg (two-step model, $k = \sqrt{|D_T|} \cdot 2$, Word2Vec) and by the competitors, with different levels of noise. The best result for a given noise amount is shown in boldface.

From the results, we can easily observe that JPReg always outperforms both LEGAL-BERT-EURLEX and BERT-PLI, independently on the amount of noise in the data. Specifically, while the impact of noise is very strong on competitors, JPReg appears very robust and, thus, adoptable in real contexts even when the amount of noise in the data is high. The significantly lower F1-score@20 results achieved by the competitors, when documents are affected by high levels of noise, can be mainly due to the higher dimensionality of their embedding space (768), with respect to that adopted in JPReg (50). Indeed, although BERT-based models exhibit very interesting results in several NLP tasks [5], their high-dimensional feature space makes them more susceptible to the curse of dimensionality on tasks based on the computation of distances/similarities [9], like in the task at hand.

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 JPReg (see Table 3).

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

Table 3. Average running time (s) for the identification of the top-n most similar paragraphs, with the two-step model and $k = \sqrt{|D_T|} \cdot 2$.

4 Conclusions

In this paper, we presented JPReg, a method to identify paragraph regularities in legal judgments. JPReg represents the documents and their paragraphs in a numerical feature space by exploiting embedding methods able to capture the context and the semantics. JPReg is based on a two-step approach, that groups similar documents into clusters and learns multiple paragraph embedding models able to represent specific topics. Finally, JPReg demonstrated to be able to identify paragraph regularities efficiently, thanks to the adopted ANNS strategy.

The accuracy and efficiency exhibited by the developed approach on real data make JPReg a useful tool in real-world scenarios, also when large collections of legal documents, possibly affected by noise, have to be analyzed.

For future work, we will exploit JPReg to provide suggestions during the preparation of new legal documents, by exploiting process mining methods. Moreover, we will evaluate the possibility to adopt transfer learning methods [15, 19] to exploit the models learned for a document cluster for the paragraphs of other, similar, clusters. Finally, we will investigate the possibility to implement JPReg as a service [20], to make it easily available to legal practitioners.

5 Acknowledgments

The authors acknowledge the support of the University of Bari Aldo Moro through the project "Horizon Europe Seeds - S34 - Hub Universitario Multidisciplinare. Processi innovativi per la digitalizzazione e la valorizzazione del patrimonio culturale scritto" (H99J21017690006).

References

1. Bernhardsson, E.: Annoy at github. https://github.com/spotify/annoy (2015)
2. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: LEGAL-BERT: The muppets straight out of law school. In: Findings of the Association for Computational Linguistics: EMNLP 2020. pp. 2898–2904 (2020)
3. Corizzo, R., Pio, G., Ceci, M., Malerba, D.: DENCAST: distributed density-based clustering for multi-target regression. J. Big Data **6**, 43 (2019)
4. De Martino, G., Pio, G., Ceci, M.: PRILJ: an efficient two-step method based on embedding and clustering for the identification of regularities in legal case judgments. Artif Intell Law (2021). https://doi.org/10.1007/s10506-021-09297-1

5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019)
6. Donghwa, K., Seo, D., Cho, S., Kang, P.: Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec. Information Sciences **477** (10 2018)
7. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: 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. p. 226–231. KDD'96 (1996)
8. Hess, S., Pio, G., Hochstenbach, M., Ceci, M.: Broccoli: overlapping and outlier-robust biclustering through proximal stochastic gradient descent. Data Mining and Knowledge Discovery **35**(6), 2542–2576 (Nov 2021)
9. Kumar, A., Makhija, P., Gupta, A.: Noisy text data: Achilles' heel of bert. In: Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020). pp. 16–21 (01 2020)
10. Kumar, S., Reddy, P.K., Reddy, V.B., Suri, M.: Finding similar legal judgements under common law system. In: Databases in Networked Information Systems. pp. 103–116. Springer Berlin Heidelberg (2013)
11. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. 31st International Conference on Machine Learning, ICML 2014 **4** (05 2014)
12. Li, W., Zhang, Y., Sun, Y., Wang, W., Zhang, W., Lin, X.: Approximate nearest neighbor search on high dimensional data - experiments, analyses, and improvement (v1.0). CoRR (2016)
13. Mandal, A., Chaki, R., Saha, S., Ghosh, K., Pal, A., Ghosh, S.: Measuring similarity among legal court case documents. In: Proc. of the 10th Annual ACM India Compute Conference. p. 1–9. Association for Computing Machinery (2017)
14. Medvedeva, M., Vols, M., Wieling, M.: Using machine learning to predict decisions of the european court of human rights. Artificial Intelligence and Law **28**(2) (2020)
15. Mignone, P., Pio, G.: Positive unlabeled link prediction via transfer learning for gene network reconstruction. In: ISMIS 2018. pp. 13–23. Springer (2018)
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems **26** (10 2013)
17. Minocha, A., Singh, N., Srivastava, A.: Finding relevant indian judgments using dispersion of citation network. In: Proceedings of the 24th International Conference on World Wide Web. p. 1085–1088. Association for Computing Machinery (2015)
18. Pio, G., Ceci, M., Loglisci, C., D'Elia, D., Malerba, D.: Hierarchical and Overlapping Co-Clustering of mRNA: miRNA Interactions. In: ECAI 2012. Frontiers in Artificial Intelligence and Applications, vol. 242, pp. 654–659. IOS Press (2012)
19. Pio, G., Mignone, P., Magazzù, G., Zampieri, G., Ceci, M., Angione, C.: Integrating genome-scale metabolic modelling and transfer learning for human gene regulatory network reconstruction. Bioinformatics **38**(2), 487–493 (09 2021)
20. Redavid, D., Corizzo, R., Malerba, D.: An OWL Ontology for Supporting Semantic Services in Big Data Platforms. In: Proc. of 2018 IEEE International Congress on Big Data, BigData Congress 2018. p. 228 – 231 (2018)
21. Shao, Y., Mao, J., Liu, Y., Ma, W., Satoh, K., Zhang, M., Ma, S.: Bert-pli: Modeling paragraph-level interactions for legal case retrieval. In: Proc. of International Joint Conference on Artificial Intelligence, IJCAI-20. pp. 3501–3507 (2020)