Searching Case Law Judgments by Using Other Judgments as a Query

No Author Given

No Institute Given

Abstract. This paper presents an effective method for case law retrieval based on semantic document similarity and a web application for querying Finnish case law. The novelty of the work comes from the idea of using legal documents for automatic formulation of the query, including case law judgments, legal case descriptions, or other texts. The query documents may be in various formats, including image files with text content. This approach allows efficient search for similar documents without the need to specify a query string or keywords, which can be difficult in this use case. The application leverages two traditional word-frequency based methods, TF-IDF and LDA, alongside two modern neural network methods, Doc2Vec and Doc2VecC. Effectiveness of the approach for document relevance ranking has been evaluated using a gold standard set of inter-document similarities. We show that a linear combination of similarities derived from the individual models provides a robust automatic similarity assessment for ranking the case law documents for retrieval.

Keywords: legal text · text similarity · full text search · NLP · document retrieval

1 Introduction: Making Case Law Search Easier

Juridical texts are widely published online by governments to make jurisdiction transparent and freely accessible to the public, organizations, and lawyers [25, 3]. As juridical data, such as case law, is published, that data should also be made easily accessible. Easier access to data leads to increased transparency, as it enables more people to access the data. Additionally, by making case law search more effortless, the workload of juridical personnel can be reduced, leading to savings in litigation costs.

This paper makes a step towards easy public access for juridical data by presenting an effective case law search method using other judgements or free text as query input. A publicly available web application for querying Finnish case law is presented to evaluate the method. The efficiency, i.e., achieving maximum productivity with minimum effort, is improved compared to traditional keyword based querying by allowing uploading of case law files. In addition, the application provides "Get similar" buttons for retrieval results enabled by a simple API that returns similar cases given an ECLI identifier [13]. This removes the need to come up with relevant keywords and allows fast exploratory search with meaningful results as queries. The API also enables use of the application's similarity computation programmatically for research purposes or for use in other applications. The prototype application has been included as a use-case application perspective in the LawSampo semantic portal [15].

Besides efficiency, document retrieval effectiveness is required to be at least satisfactory in order to improve overall document retrieval. The main concern in retrieval effectiveness is the ranking of retrievable documents based on relevance to a query [10, 23]. Using the assumption underlying vector space models, i.e., that "the relevance of a set retrieved documents to a query is approximately equal to similarity between the query and documents in retrieved set." [14], a desirable ranking can be obtained by sorting computed correlations of texts' vector representations. The vector representations are referred to as "embeddings", since the texts are embedded into a vector space. Our application combines traditional word frequency based text embedding models with newer neural network based models to provide meaningful textual similarity rankings that are able to take synonyms and other word relations into account.

In the following, the method and its prototype implementation are first described. After this, evaluation results of the underlying methods are presented. In conclusion, contributions of our experiments and related work are discussed.

2 The Finnish Case Law Finder Application

Data The Finnish case law corpus for the application is provided by the Finnish Ministry of Justice as part of Semantic Finlex data service[1] [24]. The Finnish case law corpus consists of 13 053 judgements from 1980 to 2019 at the moment. The case law texts contain references to the laws that are applied in giving the legal decisions. This helps automatic similarity computation, since judgments that have one or more applied laws in common inherently convey that they are meaningfully similar to each other. However, the laws appear in text in either abbreviated or in their full form making their identification difficult. To harmonize the texts we use regular expressions to expand the abbreviations as a preprocessing step. Another hindrance for embedding models is word inflections. Finnish language is agglutinative, which causes words to often appear in multiple forms. This leads to a large vocabulary and small frequencies for different words making it difficult to automatically infer relationships between the words with limited data. We reduce the effect of word inflections on embedding models by using LAS [19] to lemmatize, i.e., to normalize inflected words to their base form, before embedding texts for similarity computation. In addition, we filter out stopwords from the case law texts as this has been shown to improve document retrieval [8, 28].

The case law data is stored in a relational database. It contains a table for documents that includes document texts, metadata, and an integer document

[1] `https://data.finlex.fi`

identifier. The identifier corresponds to the document's index in training data for embedding models and is used to retrieve documents. The database also includes tables for users and similarity ratings to enable users to rate document pair similarities within the application. User-rated similarities are used to evaluate the application's effectiveness.

Similarity Computation Our application ranks documents for retrieval by sorting similarity values that are obtained by computing the correlation of the texts' vector representations. This is similar to the vector space model [27] that remains widely used [6]. We chose the standard method [20], cosine similarity as the application's correlation measure for text embedding similarity.

For embedding generation, we selected four models. Two of these are bag-of-words models, i.e., word-frequency based models, namely TF-IDF [30] and LDA [7]. The other two models, Doc2Vec [18] and Doc2VecC [12], represent more modern text embedding methods: they are extensions to the word embedding neural network model Word2Vec [22] that is able to map words' semantic meanings close to each other when trained with large amounts of texts. Like Word2Vec, Doc2Vec and Doc2VecC are neural networks that learn vector representations by learning to predict either missing words from context or context words given a single word.

As our models are different in nature, we created a weighted ensemble of the models to improve upon the individual model's effectiveness in producing text embeddings for ranking. Multi-co-training TF-IDF, LDA, and Doc2Vec has been shown to outperform the individual models in topic classification [16]. However, unlike topic classification, our task enables us to use a simpler approach to create ensembles of the models. Our goal is to infer real-valued similarities from texts instead of classifying the texts. Hence, we construct our ensembles models with minimal effort by computing weighted averages over the cosine similarities from the individual models' embeddings.

We obtain weights for similarities from different models' embeddings using linear regression presented in equation (1)

$$y = x_1\beta_1 + .. + x_n\beta_n + \epsilon = \boldsymbol{x}^T\boldsymbol{\beta} + \epsilon, \tag{1}$$

where $\epsilon \in \mathbb{R}$ is an error term denoting disturbance in the linear relation. Linear regression assumes its inputs $\boldsymbol{x} \in \mathbb{R}^n$ are linearly related to an observed variable $y \in \mathbb{R}$. In our case, $\boldsymbol{x}$ contains similarity values for a document pair computed from embeddings given by the individual models and y is a ground truth human assigned similarity value for the pair of case law documents.

Full Document as a Query The main goal of the application is to enable efficient and precise search with full texts. Text documents, however, come in various formats. For instance, the user might have a case law text in print or a file in some text format such as XML or plain text. Reading text content from a text file is straightforward, but analyzing a photographed image of printed text, or a PDF with text as image, requires a technique called optical character recognition (OCR). Thus, we included the Tesseract OCR [29] application in our web application to enable querying case law with photographed texts. Tesseract

OCR was chosen for the task because it is an open source OCR system that has a well performing pre-trained model for Finnish text, comprehensive documentation, and the possibility of retraining the model further. Although having a model for Finnish out-of-the-box, Tesseract OCR was not directly implemented into the software. Instead, it was first retrained to include letters “Å”, “å” and the section sign “§”, which were not included in the Tesseract OCR’s readily available Finnish text model.

Figure 1 depicts the end-user interface of the application, *Semantic Finlex case law finder*. The user is able to input a text document as a query to Finlex case law either by uploading a file or by writing text directly to the form. The query document is seen on the sub-window “Document content”. Supported file formats for uploading documents are plain text, XML, PDF, and with Tesseract OCR, image formats, such as JPEG or PNG. The text extraction mode to be used can be selected by the drop-down menu on the right bottom corner. The search form also allows the user to choose the algorithm that ranks the documents by using the drop-down menu on the left bottom in Figure 1. Here the method “Ensemble” is selected. Ranking with some algorithms may work better than others for certain topics, or depending on what kind of relatedness is preferred. Also the preferred result size can be specified.

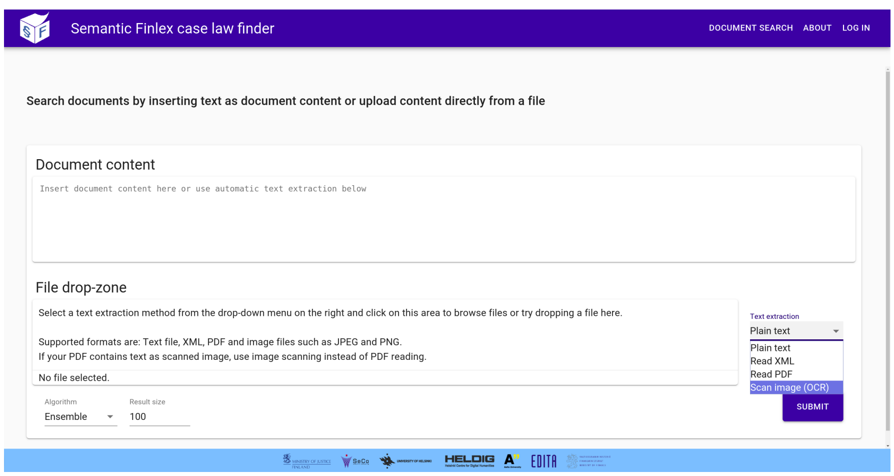

Fig. 1. Semantic Finlex case law finder application document search.

Document Ranking Once a document is submitted in the document search form, it is sent to the application back-end that handles document ranking. The back-end provides a simple API for retrieving case law documents. The query document is sent to the API via a HTTP/POST request where an embedding model is specified in the requested URI. An optional parameter n is provided to limit the number of retrieved documents, since sending the results via HTTP causes a bottleneck in retrieval time. The query document is pre-processed to the same format as models' training data, and the formatted query text is given to the model as input. The model transforms its input into a vector, and cosine similarity values are computed between the query's vector representation and all case law documents' vector representations in the underlying database. Then, all document ids are sorted by the computed similarities, and the top n ranked documents are retrieved from the database and returned in JSON-format. The query retrieval and processing is illustrated as a graph in Figure 2.

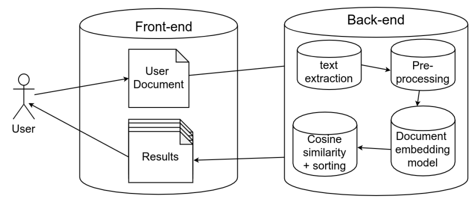

Fig. 2. Semantic Finlex case law finder application architecture overview.

The ranked cases are shown to the user as a list of expandable panels that have the case law identifier and keywords as a panel's title, as depicted in Figure 3. Similarity rating is shown by default to give insight on how the similarities change. This allows the user to see from the values whether there are likely relevant results left to view. The result items also contain the button "GET SIMILAR" for quickly querying case law related to a result document. This is intended as a helpful measure when the user does not have a certain query document, but rather wants to search the case law corpus exploratively. The retrieved results may also be further filtered by traditional methods, e.g. by using exact phrase match or filtering by court.

The created application and its ranking models are intended to work as generally as possible. While the application is created for the specific domain of Finnish case law, there is little in the implementation, besides the trained models, that restricts the application to be used with other corpora. Abbreviation expansion is the only element that ties the textual context to the Finnish language or juridical terminology. This suggests that the performance can be generalized to texts from other linguistic domains as well as other languages.

The models in the application leverage lemmatization as it was deemed beneficial in model evaluation. Additionally, by effectively normalizing word inflections, lemmatization also helps automatic inference of natural sentence queries. Thus, we performed an additional free text query test on our working application with short natural language queries to see how it would manage the task although the models are optimized for full document search. A working example is the query "törmäsin autoon" (I collided into a car). Without lemmatization, the inflected word "törmäsin" (I collided) would be non-existent in the training data. This would prevent even the neural net models from inferring that the user

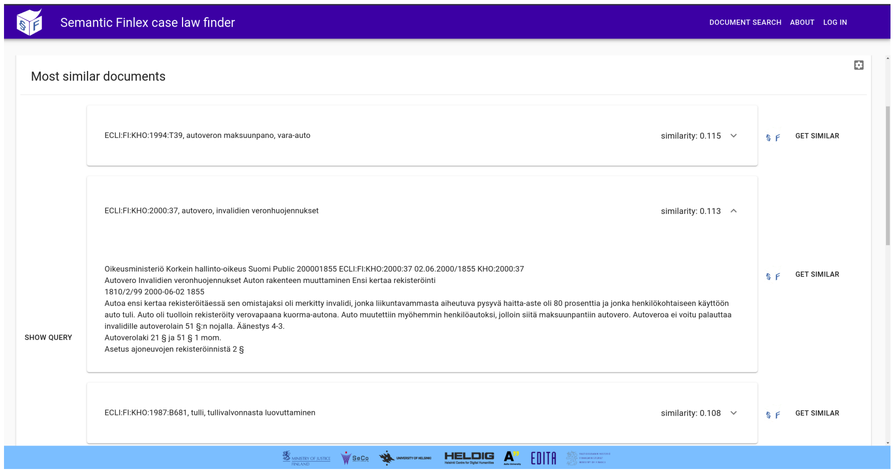

Fig. 3. Document search results in Semantic Finlex case law finder.

is inquiring about collisions. As a result of the free text query experiment, we find that the example query accurately returns cases concerning cars and traffic accidents. However, we did not perform any extensive free text query testing as this was not our primary objective.

3 Evaluation

Method We assessed ranking effectiveness of the four text embedding models that are used in the application, namely of TF-IDF, LDA, Doc2Vec and Doc2VecC. Additionally, we tested using word vector averages of Word2Vec as text embeddings. The embedding effectiveness is evaluated by comparing inter-document similarities computed from the embeddings against gold standard similarities. For our performance measures, we use Pearson correlation, Spearman rank order correlation and mean squared error. We tested different preprocessing methods effect on the models. Most of the models contain manually assignable hyperparameters that we tuned using the gold standard. For preprocessing steps, we tested the effects of lemmatization and stopword removal. Additionally, query expansion was tested with the word-frequency models TF-IDF and LDA using OIKO[2], an ontology of Finnish legal terms, and a Finnish ontology collection KOKO[3] to find synonyms, hypernyms and hyponyms for document words.

Gold Standard Labels

As ground truth data for ranking effectiveness evaluation, we use manually labeled, i.e. gold standard, similarities for selected document pairs. However, gold standard similarity scores are strenuous to obtain and can be the most

[2] https://finto.fi/oiko/en/
[3] https://finto.fi/koko/en/

resource consuming step in the creation of an information retrieval system [26]. In addition, determining how to judge document similarity might not be obvious for an annotator regardless of their expertise. Probably due to this, people tend to have different opinions on similarity [1, 2]. Therefore, it is reasonable to devise an intelligent system that is easy to use, and which leverages preferably more than one annotator to acquire similarity labels for a gold standard set.

To alleviate the manual similarity labelling process, we incorporate comparing and evaluating case law similarity within our self-built case law finder web application. Labeling inter-document similarity within the application is made possible by providing an optional sign up and login, which allows user to submit a similarity assessment for a query result. We include sign up and login requirement to select verified assessors from others, as well as to exclude thoroughly inconsistent similarity labels, i.e., random or seemingly dishonest submissions.

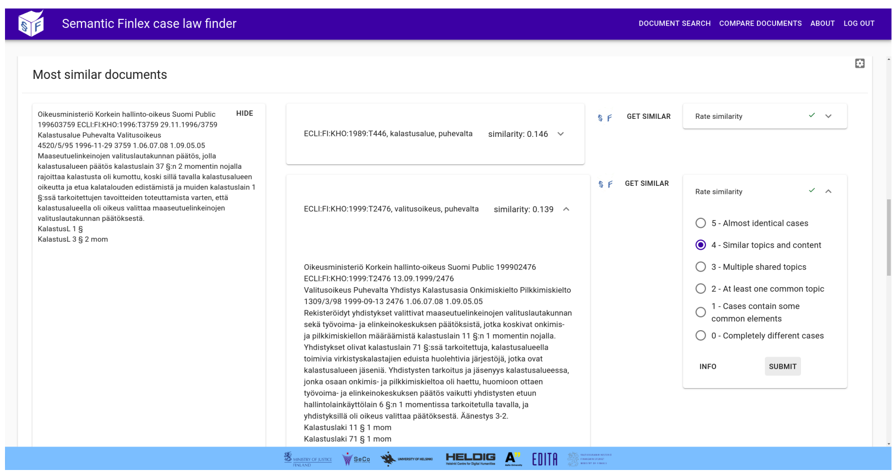

Fig. 4. Rating similarity in the Finnish case law finder application. Green tick indicates a query-result pair similarity has been rated.

Similarly to SemEval's methodology for acquiring ground truth similarities [1, 1, 21, 11], we use a 0–5 scale for similarity: Almost identical (5); Similar topics and content (4); Multiple shared topics (3); At least one common topic (2); Some common elements (1); Completely different (0). Documents for the gold standard set of inter-document similarities were selected by choosing seven "seed" documents of different topics including crime, accidental homicide, nature preservation, and name change. Then the application was used to search for documents similar to the seeds and the first results were rated. The final similarity rating set obtained includes 138 distinct ratings in total. Due to difficulty in acquiring expert labels (likely a result of lack of advertising or not providing a materialistic reward for rating), 129 of the labels are assigned by the first author of this paper, while nine are given by a volunteering law student.

As the gold standard is constructed mostly by a single person and not someone more familiar with case law, or better yet, multiple such people, our ground truth for evaluating Finnish case law ranking leaves plenty of room for improvement. However, in our defence, 2017 SemEval task [11] in multi-lingual text similarity uses 250 pairs for each language, which are either constructed by a single expert or non-experts via crowd-sourcing. Additionally, regarding the gold standard set size, Campr et al. [9] had three annotators to manually label only 150 pairs of summaries in order to compare the accuracy of various similarity computation models.

Results Consistently with all our selected assessment measures, Doc2Vec and LDA perform the best of the individual models with correlation measures between 0.60 and 0.65. TF-IDF and Doc2VecC also produce decent correlations with values below 0.6 but clearly above 0.5 for both. For Word2Vec averages, the results are not promising as its performance was far below the other models. While there are differences between the individual model performances, they are not clear enough to provide conclusive evidence for supremacy of the best performing individual models, especially when taking into account our inextensive validation set. However, both mean ensemble and linear regression ensemble outperform the individual models significantly. This shows that leveraging multiple models is recommendable when computing similarities for ranking using embedding models although it requires a bit of extra work to train all the models. The results are shown in Table 1. Regarding preprocessing steps, lemmatization is found useful for all models. For stopword removal, we found initially mixing results for its usefulness. However, after keeping the stopwords "yli" (over) and "ei" (no) in the texts while removing other stopwords, we found that stopword removal was beneficial for all models. LDA benefited from query expansion for training data while TF-IDF was unaffected.

Model	Pearson	Spearman	MSE	Embedding size	Variant	Window
TF-IDF	0.57 (0.42)	0.56 (0.50)	0.85 (1.17)	-	-	-
LDA	0.62 (0.46)	0.60 (0.48)	0.76 (1.07)	300	-	-
Word2Vec	0.42 (0.54)	0.41 (0.52)	1.16 (0.93)	900	C-BoW	10
Doc2Vec	0.64 (0.48)	0.64 (0.46)	0.71 (1.03)	500	D-BoW	5 / 10
Doc2VecC	0.56 (0.53)	0.53 (0.52)	0.89 (0.94)	700	C-BoW	10
Mean ensemble	0.70 (0.62)	0.69 (0.62)	0.61 (0.76)	-	-	-
LinReg ensemble	0.75 (0.70)	0.74 (0.70)	0.5 (0.6)	-	-	-

Table 1. Evaluation results. Correlations and mean squared error between gold standard similarities and embeddings' cosine similarity values for best hyperparameters and preprocessing steps. Results for embeddings from lemmatized texts are without brackets and ones without lemmatization in round brackets. Optimal embedding size, model variant window size and random sampling rate hyperparameters are shown for machine learning models where applicable. The slash symbol "/" between two options denotes the options resulting in equal performance for model evaluation.

Since our best performing model is linear regression ensemble, which computes weights for each individual model, we tested the validity of the weights using 5-fold cross-validation. The results, depicted in Table 2, show that for the test set, linear regression ensemble performs similarly to the mean ensemble. This suggests that more ground truth data is required to optimize the weights, but also that fine-tuning the weights might not carry much importance. We further analysed the weights to see which models are deemed the most important by the regression. By examining the regression weights in Table 3, we see that Doc2Vec contributes to approximately a third of the ensemble's similarity score. This implies that neural networks are a bigger factor in the ensembles success than the more traditional bag-of-words models. On the other hand, according to the weights, Word2Vec contributes nothing in addition to its two derivatives, and thus, is not needed in the ensemble.

Model	Pearson	Spearman	MSE
TF-IDF	0.47	0.37	1.06
LDA	0.58	0.51	0.83
Word2Vec	0.44	0.40	1.13
Doc2Vec	0.58	0.58	0.84
Doc2VecC	0.59	0.51	0.81
Mean Ensemble	0.68	0.65	0.64
LinReg Ensemble	0.68	0.65	0.64

Table 2. Average test set correlations and mean squared error between gold standard similarities and embeddings' cosine similarity values for 5-fold cross validation.

TF-IDF	LDA	Doc2Vec	Doc2VecC	Word2Vec
0.16	0.12	0.66	0.06	0.00

Table 3. Model weights for best performing ensemble

4 Contributions and Related Work

We present a public web application for efficient and effective retrieval of Finnish case law documents. The application improves upon traditional document retrieval efficiency by introducing the possibility to use documents of various formats as the query text. The general idea of searching documents similar to a description in natural language text is not new: it has been applied in situations where rich textual target documents are available, formulating the query is challenging in terms of keywords, and there is a similar document available to be used as the model to be matched. This is often the case with legal data, and there are commercial systems on the Web, such as Casetext[4] Fastcase[5], for searching legal documents using other documents as a query. The idea has been applied also in, for example, patent search[6] where a patent application or

[4] http://casetext.com
[5] http://fastcase.com
[6] Cf., e.g., http://www.acclaimip.com/

a description of it can be used for checking out whether the idea has already been patented. However, we have not been able to find research publications describing the benefits of providing easier case law search by making querying easier using documents, or about public freely available applications with such goals. Our work is an attempt to fill this gap, and to create an open source implementation and data for the task (CC-BY-4.0).

Our presented application's effectiveness is based on combining existing text embedding models for similarity computation. We evaluated TF-IDF, LDA, Doc2Vec, Doc2VecC and Word2Vec averages as embedding models for retrieval ranking against gold standard similarities. Our results show that a linear regression ensemble and also a simple mean weighted ensemble are more powerful than individual models. Based on our results, we propose using different types of models to compute inter-document similarity although this requires this the extra work to train all of them. Additionally, provided that one has a gold standard set, we suggest using linear regression or other learning method to learn optimal weighting for the models. However, we must note that our gold standard set is small and thus different results, especially for individual models, are possible with more evaluation data.

Text embeddings by neural networks has gained attention in recent work on information retrieval [5] and textual semantic similarity [21, 11]. However, the focus in semantic similarity computation has been on short texts, such as keyword queries or sentences. In contrast, the Finnish case law data used in this work consists of variously sized documents, the longer ones containing over 10 000 words. As for the legal context, neural network embeddings have been leveraged in mildly related cases. For instance, Ash et al. [4] analyse judges' relations and judicial reasoning by examining spacial relationships between case law embeddings of different judges' verdicts. Moreover, closely related to case law retrieval, Landthaler et al. [17] have used word embeddings to enhance retrieval of EU Data Protection Directives.

Acknowledgements Thanks for collaborations to Aki Hietanen, Saara Packalen, Tiina Husso, and Oili Salminen at Ministry of Justice, Finland, to Minna Tamper and Jouni Tuominen at Aalto University and University of Helsinki, and to Jari Linhala, Arttu Oksanen, and Risto Talo and at Edita Publishing Ltd. This work is part of the Anoppi project funded by Finnish Ministry of Justice[7].

References

1. E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo. * SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, volume 1, pages 32–43, 2013.
2. E. Agirre, M. Diab, D. Cer, and A. Gonzalez-Agirre. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on*

[7] https://oikeusministerio.fi/en/project?tunnus=OM042:00/2018

Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, pages 385–393. Association for Computational Linguistics, 2012.
3. E. M. M. T. A. H. Arttu Oksanen, Jouni Tuominen and E. Hyvönen. Semantic Finlex: Finnish legislation and case law as a linked open data service. In *Proceedings of Law via the Internet 2018 (LVI 2018), Knowledge of the Law in the Big Data Age, abstracts*, pages 212–228, October 2018.
4. E. Ash and D. L. Chen. Case vectors: Spatial representations of the law using document embeddings. *Social Science Research Network*, 2018.
5. M. Basu, S. Ghosh, and K. Ghosh. Overview of the fire 2018 track: Information retrieval from microblogs during disasters (irmidis). In *Proceedings of the 10th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE'18, pages 1–5, New York, NY, USA, 2018. ACM.
6. J. Beel, B. Gipp, S. Langer, and C. Breitinger. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17(4):305–338, Nov 2016.
7. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
8. T. Brants. Natural language processing in information retrieval. In *Proceedings of the 14th Meeting of Computational Linguistics in the Netherlands*, pages 1–12, 2004.
9. M. Campr and K. Ježek. Comparing semantic models for evaluating automatic document summarization. In *International Conference on Text, Speech, and Dialogue*, pages 252–260. Springer, 2015.
10. Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking SVM to document retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 186–193. ACM, 2006.
11. D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*, 2017.
12. M. Chen. Efficient vector representation for documents through corruption. In *5th International Conference on Learning Representations*. OpenReview.net, 2017.
13. Council of the European Union. Council conclusions inviting the introduction of the European Case Law Identifier (ECLI) and a minimum set of uniform metadata for case law. In *Official Journal of the European Union, C 127, 29.4.2011*, pages 1–7. Publications Office of the European Union, 2011.
14. V. N. Gudivada, D. L. Rao, and A. R. Gudivada. Chapter 11 - information retrieval: Concepts, models, and systems. In V. N. Gudivada and C. Rao, editors, *Computational Analysis and Understanding of Natural Languages: Principles, Methods and Applications*, volume 38 of *Handbook of Statistics*, pages 331 – 401. Elsevier, 2018.
15. E. Hyvönen, M. Tamper, A. Oksanen, E. Ikkala, S. Sarsa, J. Tuominen, and A. Hietanen. LawSampo: A semantic portal on a linked open data service for Finnish legislation and case law. In *Proceedings of ESWC 2020, Poster and Demo Papers*. Springer, 2020. In press.
16. D. Kim, D. Seo, S. Cho, and P. Kang. Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec. *Information Sciences*, 477:15–29, 2019.
17. J. Landthaler, B. Waltl, P. Holl, and F. Matthes. Extending full text search for legal document collections using word embeddings. In *JURIX*, pages 73–82, 2016.

18. Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196, 2014.
19. E. Mäkelä. LAS: an integrated language analysis tool for multiple languages. *The Journal of Open Source Software*, 1(6), Oct 2016.
20. C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*, chapter 6. Cambridge University Press, New York, NY, USA, 2008.
21. M. Marelli, L. Bentivogli, M. Baroni, R. Bernardi, S. Menini, and R. Zamparelli. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 1–8, 2014.
22. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.
23. E. Nalisnick, B. Mitra, N. Craswell, and R. Caruana. Improving document ranking with dual word embeddings. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, pages 83–84, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.
24. A. Oksanen, J. Tuominen, E. Mäkelä, M. Tamper, A. Hietanen, and E. Hyvönen. Semantic Finlex: Transforming, publishing, and using finnish legislation and case law as linked open data on the web. In G. Peruginelli and S. Faro, editors, *Knowledge of the Law in the Big Data Age*, volume 317 of *Frontiers in Artificial Intelligence and Applications*, pages 212–228. IOS Press, 2019. ISBN 978-1-61499-984-3 (print); ISBN 978-1-61499-985-0 (online).
25. M. Opijnen, G. Peruginelli, E. Kefali, and M. Palmirani. On-line publication of court decisions in the eu: Report of the policy group of the project 'building on the european case law identifier'. *Available at SSRN 3088495*, 2017.
26. T. Qin, T.-Y. Liu, J. Xu, and H. Li. Letor: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval*, 13(4):346–374, 2010.
27. G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, Nov. 1975.
28. J.-H. Shin, M. Abebe, C. J. Yoo, S. Kim, J. H. Lee, and H.-K. Yoo. Evaluating the effectiveness of the vector space retrieval model indexing. In *Advances in Computer Science and Ubiquitous Computing*, pages 680–685. Springer, 2016.
29. R. Smith. An overview of the tesseract ocr engine. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 02*, ICDAR '07, pages 629–633, Washington, DC, USA, 2007. IEEE Computer Society.
30. K. Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.