Extending Full Text Search for Legal Document Collections using Word Embeddings

Jörg LANDTHALER [a], Bernhard WALTL [a], Patrick HOLL [a] and Florian MATTHES [a]
[a] *Software Engineering for Business Information Systems, Department of Informatics, Technical University of Munich, Germany*

Abstract. Traditional full text search allows fast search for exact matches. However, full text search is not optimal to deal with synonyms or semantically related terms and phrases. In this paper we explore a novel method that provides the ability to find not only exact matches, but also semantically similar parts for arbitrary length search queries. We achieve this without the application of ontologies, but base our approach on Word Embeddings. Recently, Word Embeddings have been applied successfully for many natural language processing tasks. We argue that our method is well suited for legal document collections and examine its applicability for two different use cases: We conduct a case study on a stand-alone law, in particular the EU Data Protection Directive 94/46/EC (EU-DPD) in order to extract obligations. Secondly, from a collection of publicly available templates for German rental contracts we retrieve similar provisions.

Keywords. information retrieval, full text search, relatedness search, recommender systems, text mining, word embeddings, EU-DSGVO, rental contracts

1. Introduction

Information Retrieval (IR) has a long and broad history, because searching text corpora for specific information is a major task, especially in regards to legal texts. This process is nurtured by the increasing availability of legal texts such as laws, judgments and contracts in digital form. An increasing amount of legal texts is available online (cf. Winkels et al. [1]). Traditional full text search finds exact matches to a given search string in a collection of texts. While this facilitates finding relevant information, discovering all relevant documents remains challenging. Depending on the use case it might be necessary to consider synonyms and related words for each word in the search query. A common approach to address this problem is the application of domain ontologies documenting relationships such as synonyms or antonyms. However, the creation and maintenance of such ontologies is often difficult and expensive.

A recent trend in natural language processing (NLP) is to use Word Embeddings rather than term frequency vectors. Word Embeddings are dense feature vectors that capture semantics of words. We contribute and explore a novel search method that leverages Word Embeddings for searching legal document collections without integrating ontologies. Our method extends traditional full text search to not only find exact matches, but

semantically close matches. For example, within the EU-DPD act a search for the string ”the controller shall document” might yield as one result the string ”the processor shall implement”. Our approach can be used to find related rights and obligations or arguments in legal documents. The proposed method is not restricted to process single text documents. It can be used to find related text passages in a collection of laws, judgments or contracts, too. Due to the fact that representing the context of legal terms is very important, the word embeddings approach is well suited for legal documents. In addition, terms within a legal domain are often used in a consistent way of understanding and interpreting. This is a beneficial characteristic for the application of our method.

The remainder of this work is organized as follows: in Section 2 we embed our work in the existing body of literature, in particular in the context of legal informatics. Next, we give an introduction into the topic of Word Embeddings in Section 3 followed by an in-depth description of our method and a brief description of our prototype in Section 4. We evaluate the applicability of our approach in Section 5 in a case study on the recently published EU-DPD privacy act and a collection of publicly available templates of rental contracts for apartments. We critically reflect our work and discuss its limitations in Section 6. Finally, Section 7 concludes this paper with a short recapitulation of our contribution, main results and future work.

2. Related Work

Improving search, exploration, and navigation in legal data has been in the focus of legal informatics ever since. The provision of novel and innovative search services is highly relevant for legal practitioners. Different approaches have been pursued, found and successfully implemented. In [2,3] Lu and Conrad describe four views of legal documents that reflect the characteristics of legal data that can be used by modern legal search engines for search and ranking algorithms. The authors differentiate between the document, annotation, citation network and user view. Depending on the concrete task at hand, each dimension can provide input for a legal search engine. The document view has the strongest focus on the particular content of a legal document, the remaining three focus on additional meta-data, such as authorship, publishing data or classifications. Constructing a citation network and reusing references between documents to derive recommendations for a given document is explored by Winkels et al. [1] in 2014. The approach determines fully-explicit references based on lexical information (regular expressions). In addition, they attempt to capture the reason for the citation. The reconstruction of citation networks has also attracted several other researchers to investigate citations throughout laws and cases [4,5,6]. Beside taking into account metadata of legal documents, the analysis of the legal document itself has been in the focus of legal informatics [7]. Alschner et al. have used q-grams to determine similarities between clauses and norms of bilateral investment treaties [8]. Grabmair et al. have adapted and implemented an analysis pipeline to find relevant semantic patterns in vaccine injury decisions [9]. They trained a system to find relevant linguistic and semantic patterns that capture legally relevant concepts and their context.

Word Embeddings have been applied on legal texts for argumentation mining: Rinott et al. [10] detect evidences for claims in debates. In particular the authors calculate the cosine similarity of all pairs of Word Vectors in a claim and candidate text fragments that

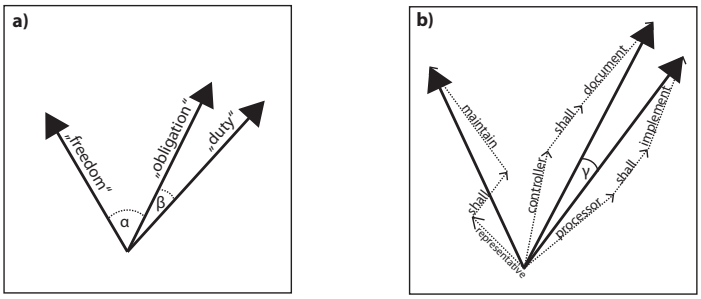

Figure 1. Simplified two-dimensional illustration of the characteristics of Word Vectors: a) Word Embeddings calculated by word2vec capture semantic relatedness, the vectors that represent the words *duty* and *obligation* are closer to each other than to the vector that represents the word *freedom*, i.e. $\alpha > \beta$, where the angle or relatedness is calculated using a similarity measure, for example cosine similarity. b) Vector addition can be imagined as the visual concatenation of arrows. Similar to a), the relatedness of arbitrary many words can be calculated, too. The bold arrows illustrate the sum of the Word Vectors of the individual words. Again, the angle $\gamma$is smaller compared to the angles among all other depicted bold arrows.

can contain evidence for the claim and rank the candidates. Naderi and Hirst [11] detect text fragments that support claims in political debates by computing similarity scores between sentences followed by a Support Vector Machine classification.

Full-text search is an out-of-the-box capability of many databases and the consideration of ontologies to reformulate search queries has been applied successfully (e.g., SwissLex [12]). Recently, Word Embeddings have been used for query-expansion in IR [13,14,15]. While query expansion is currently a more performant technique, our approach is more related to full-text search and implicitly performs query expansion. The extraction of rights and obligations from privacy laws has been investigated extensively by Kiyavitskaya et al. [16], [17] on the HIPAA[1]. However, most approaches to extract rights and obligations from privacy laws rely on rule-based techniques.

3. Word Embeddings

Word Embeddings are a neural natural language model, initially presented by Hinton et al. [18] in 1986. Due to new efficient methods of calculating Word Embeddings, started by Mikolov et al. [19] in 2013, Word Embeddings for several gigabytes of text data can be calculated within hours. This technology recently gained a lot of attention in the natural language processing community. Word Embeddings are an input feature transformation for text processing. Traditional sparse vector representations (bag-of-words) consume either a lot of memory or require specialized vector operations for sparse vectors and typically drop the word order. While Word Vectors are still considered a bag-of-words approach (addition is commutative), Word Vectors do encode the general context of words in dense vectors with manually chosen size. Mathematical operations (vector addition, multiplication) can be carried out on the vectors while preserving their inherent semantic characteristics.

Word Embeddings capture semantic characteristics. Mikolov et. al [20] show that Word Vectors trained on fictional English literature capture semantic relationships among

[1] Health Insurance Portability and Accountability Act, United States

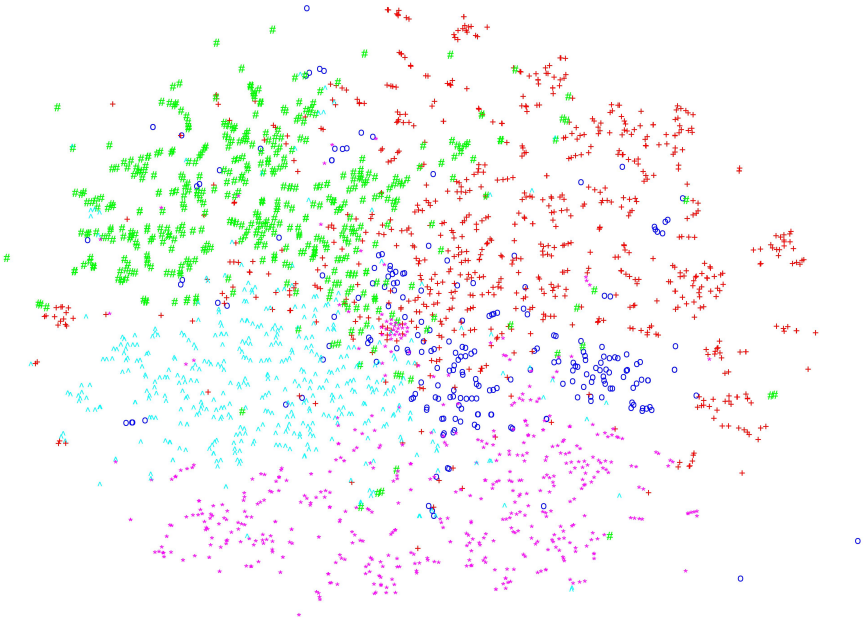

Figure 2. Clustering of German Civil Code (GCC) norm vectors. The norms of the books form clusters. Therefore, all word vectors of a norm are summed up and a t-SNE two-dimensional embedding of the resulting norm vectors is calculated. We mark and color the norms in the plot according to the book they belong to: Book 1 (o, blue), Book 2 (+, red), Book 3 (#, green), Book 4 (*, magenta), Book 5 (^, cyan). Note that the GCC as a training set for word2vec is compared to typical training sets for the algorithm very small and the clamp technique used in the GCC is clearly visible. Best viewed in color.

words. We illustrate such semantic relationships encoded in Word Vectors trained on the EU-DPD privacy act in Figure 1 a). Several approaches exist to exploit these relationships also on multiple words ranging from sentences to whole documents, e.g. paragraph vectors [21]. The most basic approach to compare strings of multiple words (tokens) is to simply sum up vectors. It is possible to average the resulting vectors by the number of tokens added up, too. In contrast to existing work - at least to our knowledge - we propose a method capable of comparing strings with an arbitrary amount of tokens, i.e. without restricting the summation of vectors to logical blocks of text. The derived relationships are depicted in Figure 1 b).

We differentiate our method to the traditional work in the machine learning and natural language processing community in the sense that we train our vectors on comparably small datasets. While it is possible for our method to train Word Vectors on much larger training sets and to apply them in our method, we found that small training sets still yield good results. To demonstrate this, we trained Word Vectors using Mikolovs original word2vec implementation[2] on the German Civil Code (GCC)[3]. We accumulated the vectors representing all words of a norm and calculated a t-SNE[4] dimensionality reduction to two dimensions on the resulting *norm vectors*, a common way to show the qual-

[2] `https://code.google.com/archive/p/word2vec/`

[3] `http://www.gesetze-im-internet.de/bgb/index.html`, version from the 30. April 2014

[4] `https://lvdmaaten.github.io/tsne/`

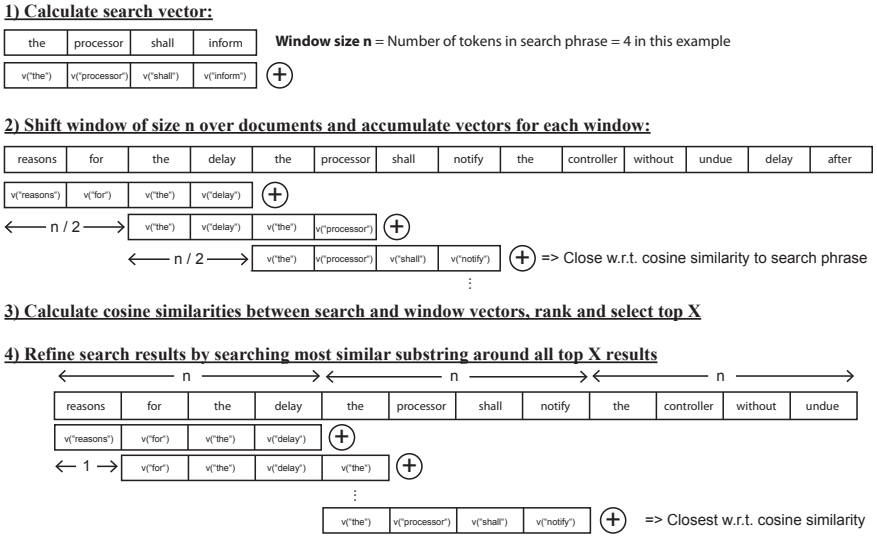

Figure 3. Illustration of how our proposed method works for searching related strings in a given text corpus that consists of one single or several different texts. v("word") denotes the corresponding Word Vector for the *word*.

ity of the calculated Word Vectors. The results depicted in Figure 2 indicate that Word Vectors calculated on a comparably small dataset work surprisingly good. The norms of each book of the GCC form clusters. In addition, the clamp technique used in the GCC is clearly visible. The clamp technique (Book 1 of the GCC contains definitions that are used in the remaining Books of the GCC) can be seen in the graph of direct references in [22]. Similarly, half of the norm vectors of Book 1 are placed in the center while the other half form a separate cluster center, i.e. half of the norms are related to other norms while the other half is not.

4. Method Description & Minimal Viable Prototype

The goal of our method is to provide a way to find not only exact matches of arbitrary length search phrases, but also to find parts in the text that might be relevant or semantically close, but differ in one or multiple words. We use Word Embeddings and the ability to sum or average vectors over multiple words.

The Word Vectors for each word of the search phrase are accumulated yielding one vector representing the search phrase (search phrase vector). For an optimal result, it would be nice to be able to compare the search phrase vector with the vectors representing every possible combination of consecutive words in a corpus of documents of arbitrary length. However, because this would be a power set of the total number of words present in the corpus, it is computationally not feasible. Thus, we use a form of subsampling. We search coarsely by moving a window of size n (the number of words in the search phrase) and restrict the search to phrases of equal length. We sequentially shift the window by n/2 over the corpus as illustrated in Figure 3. We then calculate the cosine similarity of the search phrase vector and all accumulated vectors. Next, we rank the vec-

tors and select the best X matches. In a subsequent step, we reconsider all matches and search in a window that has three times the number of words in the search phrase around the best matches and shift the window word by word. Each shift yields another vector that is compared to the search phrase vector w.r.t. cosine similarity and the best match is selected. Once a match is selected by the first iteration over the corpus we subsequently select the best match in an area around the match. We drop matches that occur at equal or close offsets in the corpus.

One possibility to train Word Vectors is to pre-train the vectors on a large corpus of legal texts. However, even across different sub-domains of the legal domain the meaning of words could differ. Since our experiments with Word Vectors on small training sets were successful (cf. Section 2, we train the Word Vectors on the corpus we want to search in. We use Mikolov's original word2vec implementation with default parameters for this algorithm, except for the number of iterations, which we set to 1000 and the minimal count of words to include them in the training, which we set to 1 to ensure that a Word Vector for all words exists. We leave the size of the vectors as a configurable parameter for our experiments, cf. Section 5. The corpus text needs to be pre-processed. We need a single-white-space separated list of words without special characters, punctuation or other white-space characters remains. It is possible though, to encode special characters, for example §with a reserved word. We assume that strings matching across existing logical borders (different documents) make few sense. Thus, we start the first step of our method for each document of a collection separately. If a document is smaller than the window size, we simply sum up the vectors of all words of a document. Missing word vectors for words of the search phrase can be a problem and so far we ignore words where no corresponding Word Vector exists or restrict the user to search by interactively selecting text in documents of the corpus.

We implemented our approach using Python[5] and Numpy[6]. It is build such that it is a service, where all operations can be consumed via a REST API implemented with Flask[7]. We decided to support a general *collections-that-contain-documents* structure so that we can map a large number of use cases easily to our system (e.g. a collection containing a single document, a collection where the documents are the norms of the GCC or a collection where documents consist of different contracts). Our prototype has been designed to provide easy exploration and evaluation of use cases rather than to maximize performance. If a document is appended to or edited in a collection, we recalculate all Word Vectors within that collection. Mikolovs word2vec implementation also outputs a vocabulary and we use this vocabulary to calculate an integer representation of all texts in a collection, because integer comparisons are much faster than string comparisons (used for the lookup table that maps words of a text to their Word Vectors).

5. Evaluation: Case Study

We show the potential applicability of our method on two different datasets: the EU-DPD act (28742 words in total, 1782 unique, without preamble) and a collection of 10 German rental contract templates (28221 words in total, 3929 unique) available on the internet.

[5] `http://www.python.org`
[6] `http://www.numpy.org`
[7] `http://flask.pocoo.org`

The datasets have different characteristics. The EU-DPD[8] has been published recently by the European Union and replaces national privacy laws. We evaluate our method for its capability to extract certain obligations for processors from the EU-DPD. The contracts contain provisions with equal or similar formulations and we show the returned results when searching for a specific provision contained in one of the contract templates. We trained the Word Embeddings on the respective dataset only with 200 dimensions and add Word Vectors without averaging.

Manual Search Results		
#O	**String**	**F**
2	the processor shall designate	✓
1	the processor shall inform	✓
1	processor shall inform the	✓
1	processor shall immediately inform	✓
1	the processor shall implement	✓
2	the processor shall notify	✓
2	the processor shall take	x
1	processor shall take the	✓
1	the processor shall publish	✓
3	the processor shall ensure	x
1	the processor shall support	✓
1	the processors shall make	x
1	the processor shall document	x
19		**11**

First 18 Prototype Search Results		
SS	**String**	**R**
1.0	processor shall inform the	✓
1.0	processor shall inform the	✓
0.94	processor shall immediately inform	✓
0.87	processor shall document the	✓
0.86	the processor shall designate	✓
0.86	the processor shall designate	✓
0.86	the initial processor shall	x
0.85	processor shall support the	✓
0.83	processor shall publish the	✓
0.83	the processor shall implement	✓
0.83	the processor shall notify	✓
0.83	processor shall notify the	✓
0.82	the controller shall inform	x
0.82	the controller shall inform	x
0.82	the controller shall inform	x
0.82	controller shall inform the	x
0.82	controller shall inform the	x
0.82	shall inform the controller	x

Table 1. Comparison of manual search and returned results of our prototype on the EU-DPD dataset for the search phrase 'the processor shall inform'. On the left we count the number of occurrences of the phrase (#O), the phrase and indicate, if all occurrences have been detected (F). On the right we show the similarity score (SS), the matched phrase and the membership to the obligations on the left (R).

From the EU-DPD we extract obligations for processors. Put in a simple way, processors are defined as persons that operate on personal data and are usually advised by a controller (data owner). We conduct a full text search on the term 'processor' yielding 174 hits. We select a subset of 19 matches, where obligations are of the form 'the processor shall *verb*'. The list of verbs is depicted in Table 1 on the left. We use the phrase 'the processor shall inform' as the search phrase in order to find similar obligations. The results are hard to put in few numbers. Table 1 shows the first 13 results returned by our prototype. 57% of expected obligations are found within the first 30 results. The others can not be found in the first 150 results. Also, the method does not only return formulations with similar words, but also often returns formulations of the form 'the controller shall'. We assume that our approach could be improved by using Word Embeddings trained on larger datasets and also from hybrid approaches with parse trees and rule-based systems.

[8] `https://www.datenschutz-grundverordnung.eu/wp-content/uploads/2016/05/CELEX_32016R0679_EN_TXT.pdf`, version from the 20 August 2016

On the publicly available set of 10 German rental contracts we show the results for an exemplary search phrase. The phrase we search for covers a common topic in rental contracts in Germany and declares that a sublease requires an approval by the landlord (dt. Vermieter): *"Die Untervermietung der Wohnung oder von Teilen der Wohnung bedarf der Erlaubnis des Vermieters."* found in one of the documents. The first 12 recommended results of our approach are listed in Table 2. The resulting list starts with the exact match and subsequently provides results from most other documents. In contrast to our search on the EU-DPD the search phrase is long. Four documents contain equal provisions on the search topic. This does not apply to all provisions in these four documents. On the one hand, some retrieved phrases are unexpected, because they govern economic use of apartments. On the other hand four different wordings of the provision searched for are found.

Doc	SS	Recommended Phrase	R
2	1.00	die untervermietung der wohnung oder von teilen der wohnungbedarf der erlaubnis des vermieters	✓
2	0.88	fur die geschafts oder gewerbeaus ubung bedarf der schriftlichen erlaubnis des vermieters der	-
4	0.86	der wohnung bedarf der mieter der vorherigen schriftlichen zustimmung des vermieters wenn er	-
7	0.84	der mieter darf die wohnung nur mit erlaubnis des vermieters untervermieten der vermieter	✓
6	0.84	der mieter darf die wohnung nur mit erlaubnis des vermieters untervermieten der vermieter	✓
8	0.84	der mieter darf die wohnung nur mit erlaubnis des vermieters untervermieten der vermieter	✓
3	0.84	der mieter darf die wohnung nur mit erlaubnis des vermieters untervermieten der vermieter	✓
9	0.82	oder die uberlassung der mietsache an dritte der zustimmung des vermieters bedarf die	✓
4	0.80	interesse einer ordnungsgemassen bewirtschaftung des hauses und der wohnung bedarf der mieter der	✓
2	0.80	pflicht des vermieters tierhaltung die tierhaltung in der wohnung ist ohne erlaubnis des	-
1	0.79	der mieter ist ohne vorherige erlaubnis des vermieters nicht berechtigt die angemieteten raume	✓
4	0.79	fur die untervermietung bedarf es der zustimmung des vermieters verweigert der vermieter die	✓

Table 2. First 12 returned results from our prototype on 10 publicly available rental contract templates for the search phrase *"Die Untervermietung der Wohnung oder von Teilen der Wohnung bedarf der Erlaubnis des Vermieters."*. A result row consists of a document identifier (Doc), the similarity score (SS), the recommended phrase and the relevance (R) of the result. Umlaute have been replaced.

6. Critical Reflection & Limitations

While our method and our prototype give rise to promising and interesting results, there are several limitations to both. Currently, a major limitation is the performance of our method. On the one hand, datasets need to have a minimal size of at least 10 to 20 pages

of text, so that word2vec can calculate sensible Word Vectors. On the other hand, the datasets cannot be too large, because the performance of the calculation of similar strings is slow in the current implementation. The calculation of the summed vectors for all windows is the major bottleneck. For a quick response, a maximum of 200 to 500 pages per collection should not be exceeded, except it is acceptable to wait for results. The performance could be improved by for example by caching, smaller vectors or projection matrices. There is no standardized quality measure for Word Vectors so far. The evaluation of search results is difficult, because of the many possibilities for human intention when searching and *Related information* is vague. Hence, it is not easy to assess whether all relevant parts of information have been found. Last but not least, we know that updating documents can become tedious, because all vectors need to be recalculated or updated.

7. Conclusion & Future Work

In this paper we present a novel method that enhances full text search in single documents or document collections to find exact and semantically related matches using Mikolovs word2vec implementation for Word Embeddings. We found that our method delivers relevant parts of laws and contracts when searching for rights and obligations in the EU-DPD and it detects similar provisions in rental contracts. Our minimal viable prototype is mainly designed for maximum flexibility with respect to different use cases and the performance of our method could be significantly improved by adapting it to the specific use case at hand.

For the future, it would be interesting to experiment with other natural language preprocessing steps in combination with Word Embeddings, for example stemming, stopword removal, part-of-speech tagging or parsing. The development of a standardized quality measure for Word Vectors could help to improve the understanding and reliability of this technology. It would be useful to improve the performance or to develop similar methods that are faster and scale better to larger document collections. Simple yet powerful ways to achieve this are the easy parallelization of the addition of Word Vectors and caching accumulated (frequently occurring) vectors. Last but not least, our method would benefit from improved ways of evaluation.

References

[1] R. Winkels, A. Boer, B. Vredebregt, and A. van Someren, "Towards a Legal Recommender System," in *Frontiers in Artificial Intelligence*, 2014, vol. Volume 271: Legal Knowledge and Information Systems, pp. 169–178. [Online]. Available: http://ebooks.iospress.nl/volumearticle/38453

[2] Q. Lu and J. G. Conrad, "Bringing Order to Legal Documents An Issue-based Recommendation System via Cluster Association," *International Conference on Knowledge Engineering and Ontology Development*, 2012.

[3] VOXPOPULII, "Next Generation Legal Search - It's Already Here." [Online]. Available: https://blog.law.cornell.edu/voxpop/2013/03/28/next-generation-legal-search-its-already-here/

[4] T. Agnoloni and U. Pagallo, "The case law of the Italian constitutional court, its power laws, and the web of scholarly opinions," in *15th International Conference on Artificial Intelligence and Law (ICAIL)*, K. Atkinson and T. Sichelman, Eds., 2015, pp. 151–155.

[5] J. H. Fowler, T. R. Johnson, J. F. Spriggs, S. Jeon, and P. J. Wahlbeck, "Network analysis and the law: Measuring the legal importance of precedents at the US Supreme Court," *Political Analysis*, vol. 15, no. 3, pp. 324–346, 2007.

[6] R. Boulet, A. F. Barros-Platiau, and P. Mazzega, "35 years of Multilateral Environmental Agreements ratifications: a network analysis," *Artificial Intelligence and Law*, vol. 24, no. 2, pp. 133–148, 2016.

[7] E. Francesconi, Ed., *Semantic processing of legal texts: Where the language of law meets the law of language*. Springer, 2010.

[8] W. Alschner and D. Skougarevskiy, "Consistency and legal innovation in the BIT universe," *Stanford Public Law Working Paper No. 2595288*, 2015. [Online]. Available: http://ssrn.com/abstract=2595288

[9] M. Grabmair, K. D. Ashley, R. Chen, P. Sureshkumar, C. Wang, E. Nyberg, and V. R. Walker, "Introducing LUIMA: An Experiment in Legal Conceptual Retrieval of Vaccine Injury Decisions Using a UIMA Type System and Tools," in *ICAIL '15: Proceedings of the 15th International Conference on Artificial Intelligence and Law*. New York, NY, USA: ACM, 2015, pp. 69–78.

[10] R. Rinott, L. Dankin, C. Alzate, M. M. Khapra, E. Aharoni, and N. Slonim, "Show me your evidence–an automatic method for context dependent evidence detection," in *Proceedings of the 2015 Conference on Empirical Methods in NLP (EMNLP), Lisbon, Portugal*, 2015, pp. 17–21.

[11] N. Naderi and G. Hirst, "Argumentation mining in parliamentary discourse," in *Proceedings of the Computational Models of Natural Argument 2016, New York, United States*, 2016.

[12] J. Erbguth and M. S. Bloch, "Neue Suche bei Swisslex," in *Tagungsband des 18. Internationalen Rechtsinformatik Symposions IRIS 2015*, E. Schweighofer, F. Kummer, and W. Hötzendorf, Eds. OCG – Oesterreichische Computer Gesellschaft 2015, vol. 2015.

[13] D. Ganguly, D. Roy, M. Mitra, and G. J. Jones, "Word embedding based generalized language model for information retrieval," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '15. New York, NY, USA: ACM, 2015, pp. 795–798. [Online]. Available: http://doi.acm.org/10.1145/2766462.2767780

[14] H. Zamani and W. B. Croft, "Embedding-based query language models," in *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, ser. ICTIR '16. New York, NY, USA: ACM, 2016, pp. 147–156. [Online]. Available: http://doi.acm.org/10.1145/2970398.2970405

[15] N. Ould Amer, P. Mulhem, and M. Géry, "Toward Word Embedding for Personalized Information Retrieval," in *Neu-IR: The SIGIR 2016 Workshop on Neural Information Retrieval*, vol. abs/1606.06991, Pisa, Italy, Jul. 2016. [Online]. Available: https://hal-ujm.archives-ouvertes.fr/ujm-01377080

[16] N. Kiyavitskaya, N. Zeni, T. D. Breaux, A. I. Antón, J. R. Cordy, L. Mich, and J. Mylopoulos, *Automating the Extraction of Rights and Obligations for Regulatory Compliance*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 154–168. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-87877-3_13

[17] N. Kiyavitskaya, N. Zeni, T. D. Breaux, A. I. Antón, J. R. Cordy, L. Mich, and J. Mylopoulos, "Extracting rights and obligations from regulations: Toward a tool-supported process," in *Proceedings of the Twenty-second IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '07. New York, NY, USA: ACM, 2007, pp. 429–432. [Online]. Available: http://doi.acm.org/10.1145/1321631.1321701

[18] G. E. Hinton, J. L. McClelland, and D. E. Rumelhart, "Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1," D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, Eds. Cambridge, MA, USA: MIT Press, 1986, ch. Distributed Representations, pp. 77–109. [Online]. Available: http://dl.acm.org/citation.cfm?id=104279.104287

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: http://arxiv.org/abs/1301.3781

[20] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119. [Online]. Available: http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf

[21] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents." in *ICML*, vol. 14, 2014, pp. 1188–1196.

[22] J. Landthaler, B. Waltl, and F. Matthes, "Unveiling references in legal texts - implicit versus explicit network structures," in *Tagungsband des 19. Internationalen Rechtsinformatik Symposions IRIS 2016*, E. Schweighofer, F. Kummer, W. Hötzendorf, and G. Borges, Eds. OCG – Oesterreichische Computer Gesellschaft 2016.