Legal Knowledge and Information Systems
A. Wyner and G. Casini (Eds.)

doi:10.3233/978-1-61499-838-9-183

Utilizing Vector Space Models for Identifying Legal Factors from Text

Mohammad H. Falakmasir [a] and Kevin D. Ashley [a,b]
[a] *Intelligent Systems Program, University of Pittsburgh*
[b] *School of Law, University of Pittsburgh*

Abstract.
Vector Space Models (VSMs) represent documents as points in a vector space derived from term frequencies in the corpus. This level of abstraction provides a flexible way to represent complex semantic concepts through vectors, matrices, and higher-order tensors. In this paper we utilize a number of VSMs on a corpus of judicial decisions in order to classify cases in terms of legal factors, stereotypical fact patterns that tend to strengthen or weaken a side's argument in a legal claim. We apply different VSMs to a corpus of trade secret misappropriation cases and compare their classification results. The experiment shows that simple binary VSMs work better than previously reported techniques but that more complex VSMs including dimensionality reduction techniques do not improve performance.

Keywords. Vector Space Models, Legal Analytics, Semantic extraction

1. Introduction

An important target of argument mining efforts in the legal field has been to extract factors from case texts. See [1, Chapter 10]. Legal factors are stereotypical patterns of fact that tend to strengthen or weaken a side's argument in a legal claim. [2, p.27].

Factors are particularly important in trade secret law. Information may qualify as a *trade secret* if it:

> is secret in the sense that it is not generally known among or readily accessible to people in the wider community that normally deal with the kind of information; has commercial value because it is secret; and has been subject to reasonable steps under the circumstances, by the person lawfully in control of the information, to keep it secret.[1]

Misappropriation consists of:

> acquisition of a trade secret of another by a person who knows or has reason to know that the trade secret was acquired by improper means; or disclosure or use of a trade secret of

[1] Trade Related Aspects of Intellectual Property Rights (TRIPS). Agreement on Undisclosed Information. Section 7: Protection of Undisclosed Information, Article 39.

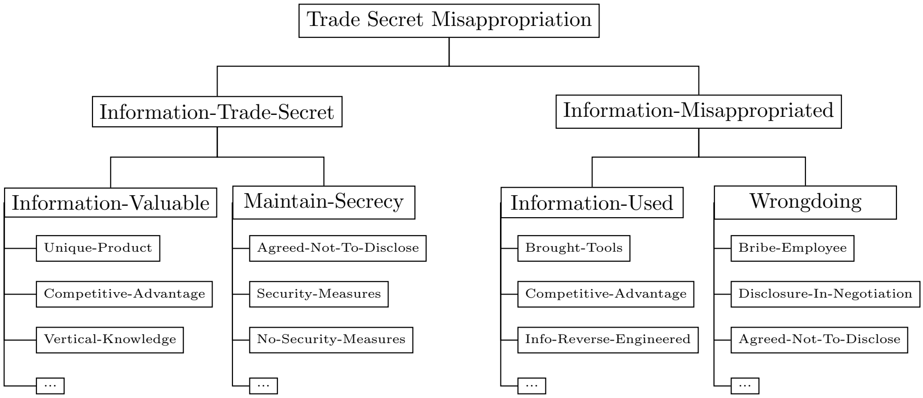

Figure 1. Example Factors from the Trade Secret Domain Model

> another without express or implied consent by a person who (i) used improper means to acquire knowledge of the trade secret ....[2]

Improper means include:

> theft, bribery, misrepresentation, breach or inducement of a breach of a duty to maintain secrecy, or espionage through electronic or other means [but not] reverse engineering, independent derivation, or any other lawful means of acquisition.[3]

A still influential secondary source of trade secret law introduced factors in an oft-cited set of guidelines for determining if information is a trade secret:

> An exact definition of a trade secret is not possible. Some factors to be considered in determining whether given information is one's trade secret are:
>
> 1. the extent to which the information is known outside of his business;
> 2. the extent to which it is known by employees and others involved in his business;
> 3. the extent of measures taken by him to guard the secrecy of the information;
> 4. the value of the information to him and to his competitors;
> 5. the amount of effort or money expended by him in developing the information;
> 6. the ease or difficulty with which the information could be properly acquired or duplicated by others.[4]

In the U.S. common law system, judges weigh the factors in a current case and explain their judgments by citing the statutes and guidelines and by making arguments based on past decisions or precedents. In modeling such case-base arguments, Ashley introduced dimensions to represent and elaborate the above factors into a set that ultimately comprised twenty-six factors, each favoring one side or the other [2]. For a complete list, see [3].

Ashley and Brüninghaus organized the claim requirements and factors into a domain model for the issue-based prediction system (IBP) [4]. Grabmair extended the model in the Value Judgment Formalism framework (VJAP) [5]. In this model (Figure 1) each factor is related to a high-level statutory requirement of a trade

---

[2] 18 U.S. Code 1839 - Definitions (5).
[3] 18 U.S. Code 1839 - Definitions (6)(B).
[4] Restatement (First) of Torts Section 757. Liability for Disclosure or Use of Another's Trade Secret. Comment b. Definition of trade secret states.

secret misappropriation claim. Each factor weighs in favor of one side or other. For example, [F6:Security-Measures] favors the plaintiff trade secret holder and indicates that it applied active measures to limit access and distribution of the information that is the property of interest. [F24:Info-Obtainable-Elsewhere] implies that the confidential information could be obtained from publicly available sources and favors the defendant, the alleged misappropriator.

For purposes of modeling, a conclusion that a factor applies in a case is based on classifying at least one sentence in the text as an instance of the factor. For example, the following sentences justify the conclusion that the associated factors apply in the *Mason* case, a trade secret dispute concerning the recipe for a drink, Lynchburg Lemonade (See [1, Figure 11.8]:)

- **F6:Security-Measures (pro-plaintiff):** He testified that he told only a few of his employees--the bartenders--the recipe. He stated that each one was specifically instructed not to tell anyone the recipe. To prevent customers from learning the recipe, the beverage was mixed in the "back" of the restaurant and lounge.
- **F15:Unique-Product (pro-plaintiff):** It appears that one could not order a Lynchburg Lemonade in any establishment other than that of the plaintiff.
- **F16:Info-Reverse-Engineerable (pro-defendant):** At least one witness testified that he could duplicate the recipe after tasting a Lynchburg Lemonade.
- **F21:Knew-Info-Confidential (pro-plaintiff):** On cross-examination Randle agreed that he had been under the impression that Mason's recipe for Lynchburg Lemonade was a secret formula.

Our main research goal is to improve the performance of automatically classifying the texts of trade secrets misappropriation cases by their applicable factors. As an initial step we asked how well vector space models (VSMs) can identify factors in the case texts (see Section 2). Using the domain model of Figure 1, once factors are identified, one could also identify the legal issues litigated in the case.

In our study, eight different VSMs plus variations learn different representations of the case texts in our corpus. Four of the VSMs are based on relatively simple binary or TF-IDF representations (see section 3). The other four employ dimensionality reduction techniques to represent case texts. We compared the representations learned by the different VSMs in terms of their results on classifying a subset of a gold standard corpus of 172 cases tagged by legal experts as to applicable factors. Initially, we hypothesized that the dimensionality reduction techniques would lead to learning VSMs that were more expressive of the underlying legal factors. Based on the results reported below, we can reject that hypothesis. Nevertheless, all of the VSMs outperformed previously-reported results in classifying case texts by legal factors.

2. Background on Vector Space Models

Statistical studies of semantics represent meaning as a probability distribution over a set of latent dimensions using the bag-of-words hypothesis or the distributional hypothesis [18]. The key idea is that if units of text have similar vectors in a term frequency matrix, they tend to have similar meaning.

Based on the bag-of-words hypothesis, word frequencies in a document indicate the relevance of the document to a search query. Given a large corpus, one can form a term-document matrix where the rows correspond to terms and the columns correspond to the frequencies of words in each document. Most of the elements of the term-document matrix are zero since most documents use only a small fraction of the whole vocabulary. The term-document matrix provides a very broad notion of meaning that is suitable for document retrieval. However, it only supports a coarse-grained measure of topical similarity [18].

Based on the distributional hypothesis, words that appear together in the same context tend to have similar meaning. The context could be a sentence, or perhaps even a fixed window of words. In general, shorter windows tend to capture syntactic features while longer windows tend to capture more semantic relations. The distributional hypothesis is the main inspiration of the recent neural network-based models for learning word vectors (word embeddings a.k.a word2vec) [18].

Vector space models of semantics represent meaning as a coordinate in a high-dimensional "semantic space". Vector representations are a common way to compute semantic similarity between arbitrary spans of text. Each context vector is a point in $|V|$-dimensional space. $|V|$, the length of the vector, is generally the size of the vocabulary. Quite often, raw term frequencies (TFs) are not the best measure of semantic similarity because word frequencies follow a skewed distribution according to Zipf's Law. An alternative measure of similarity between documents is TF-IDF. The TFs are often weighted by the inverse document frequency (IDF) to give a higher weight to rare words that occur only in a few documents [18]. The nature of the vectorized representation allows documents to be compared in terms of semantic similarity using any of the standard similarity or distance measures available from linear algebra (e.g., cosine similarity or Euclidean distance).

One can also apply various dimensionality reduction techniques, such as singular value decomposition (SVD), non-negative matrix factorization (NMF), and Latent Dirichlet Allocation (LDA) [18]. These methods can essentially be thought of as a way to cluster words along a small number of *latent* semantic dimensions that are automatically learned from a low-rank approximation of the term-document matrix. In fact, Latent Semantic Analysis (LSA) is a low-rank approximation of the term-document matrix using SVD, and both LDA and NMF has been successfully applied in the literature for topic modeling [18].

3. Data and Methods

For this study, we compiled a corpus of trade secret misappropriation cases by scraping the texts of 1,600 federal and state opinions retrieved from the CourtListener website[5] that contain references to two particular sources of legal rules: (1) the Restatement of Torts section 757, comment b (1939) ("RT757") and (2) Uniform Trade Secrets Act (1985) ("UTSA"). We also used a gold-standard corpus of 172 cases from the HYPO, CATO, SMILE, and VJAP programs (VJAP corpus) whose sentences legal experts labeled according to the 26 trade secret factors. Table 1 provides summary statistics of these corpora. The totals correct for the fact that some of the cases cite both references.

[5]https://www.courtlistener.com/

Table 1. Summary statistics of the available corpora.

Corpora	# Cases	# Sentences	# Terms	# Verbs
Restatement of Torts 757	509	108,186	36,454	26,630
Uniform Trade Secret Act	1,213	226,556	52,232	36,973
VJAP (based on HYPO, CATO, IBP)	179	26,296	19,327	13,884
Total (Unique Cases)	1,600	334,742	62,472	44,559

The performance of machine learning methods depends heavily on the choice of data representation (or features) to which they are applied. Domain knowledge can be an important resource for designing effective text representations. Feature-engineering is labor-intensive, however, and domain models evolve over time. Ideally, a representation would capture the underlying distribution of the data and automatically account for the evolution of these abstractions. Our goal in this project is to see how far one can go without feature-engineering. At the same time, we remain open to applying techniques to efficiently incorporate domain knowledge where feasible, a task for future work.

We designed our experiments as a four-step pipeline. The first step (preprocessing) includes tokenization, part-of-speech tagging, and extracting main verbs of the sentence. In the second step (vectorization) we learn multiple vector representations for the opinions in the case base. Then, we form the term-document matrix for the corpus based on the bag-of-words hypothesis.

Our binary VSM models, bag-of-words (BOW) and bag-of-verbs (BOV) represent each document using one-hot encoding, that is, with one Boolean column for each category. Since many factors correspond to parties' actions, we created the bag-of-verbs version of the term-document matrix [19]. We use a modified form of the verb by concatenating the immediate conjunct of the verb according to the dependency parse results. For example in this form, we have separate tokens for the verb "disclose" including, disclosed, not_disclosed, not_to_disclose, have_disclosed, etc. This way of representing verbs is different from the forms used in the topic modeling literature that often uses the stemmed version of the verbs ("disclos" for all of the above forms) and removes the conjuncts as stop words. The main reason is that in the legal context in general and in considering the factors in particular, the verb tense and the modals play a pivotal role in the fact finding process of the decision maker and should not be mapped into the same dimension (considered as the same "token") in the feature space.

The next two VSMs, TF-IDF (Terms) and TF-IDF (Verbs) are standard TF-IDF transformations on the term-document or verb-document matrix. We use document frequencies and raw counts as a filter to remove case-specific information. Since our evaluation set (VJAP corpus) only contains 172 carefully selected opinions (not a random sample) we use the larger corpus of 1,600 scraped opinions to calculate the counts and apply the TF-IDF weights.

In the third step (transformation) we apply four widely used VSM models to reduce the dimensionality of our representations and infer latent dimensions. Latent Semantic Analysis (LSA) and Non-negative matrix factorization (NMF) are two of the numerical approaches for transforming documents into a semantic vector spaces. Latent Dirichlet Allocation (LDA) and Hierarchical Dirichlet Process (HDP) are probabilistic alternatives for inferring latent dimensions.

In the fourth step (evaluation) we used the VSM representations of the documents from the VJAP corpus in a supervised classification framework to predict the factors and to investigate the association of each VSM with the targeted labels, the legal factors. We evaluate all of the representations (i.e., BOW, BOV, TF-IDF (Terms), TF-IDF (Verbs), LSA, NMF, LDA, and HDP) by comparing their resulting classification results.

4. Experiments

A key aspect of creating an expressive VSM model is choosing the right size of representations K for different applications (datasets). This is provided as a parameter of the model (similar to parameter K of k-means clustering). Assuming that there is a finite number of legal factors one can assume that each case is a point in a 26-dimensional space. The classifier is finding a *surface* that has only points with positive labels on one side and points with negative labels on the other side. There are numerical methods for identifying the right K; however, in this study we experiment with different values for K to investigate the effect of the size of representation for the task at hand.

We start with a term-document matrix ($X_{m \times n}$) with real-valued, non-negative entries (TF-IDF weights). Among the various ways of learning document representations, this paper focuses on low-rank approximation of the term-document matrix in the form of:

$$X_{m \times n} = W_{m \times r} \times H_{r \times n} \qquad r < \min(m, n) \tag{1}$$

The term-document matrix X is factorized into two smaller matrices, W as a document archetype that encapsulates the intensity (weight) of each term in the feature space and H that represents the projection of each document into that feature space. m is the number of terms in our corpus, n is the number of documents, and r is the dimension of our representation (feature space).

For the bag-of-words (BOW) and bag-of-verbs (BOV) VSMs, the vectorized representation of each case is a $|V|$ dimensional one-hot vector ($[0, 1, 1, 0, 0]_{|V|}$) that is created by considering terms and verbs. The TF-IDF (Terms) and (Verbs) VSMs use a standard TF-IDF weighting with n-grams (n=1, 2, 3). We filter out terms that appeared in more than 90% of the documents or fewer than 5 times throughout the corpus. We also report the results of four widely used VSM models (LSA, NMF, LDA, HDP) and experimented with different Ks to find the optimum number of dimensions. The HDP model is a non-parametric method and does not require the number of dimensions to be specified in advance. All of the experimental models can be considered as a relaxed form of k-means clustering, with columns of the W representing the cluster centroids and rows of the H indicating cluster membership (weights) for each document. As a result, the output of our VSMs are r dimensional vectors for each document.

For evaluation, we used a Support Vector Machine (SVM) with a linear kernel in a multi-label (One-vs-Rest) classification framework. We train a binary SVM classifier for each factor without performing any parameter optimization. Although one can tune the C parameter of the SVM classifier to increase the recall at the expense of lower precision [16], we decided to use F1 as our evaluation

Table 2. Experimental Results

#	VSM / Results	Precision (mi/ma)	Recall (mi/ma)	F1 (mi/macro)	#Features
1	BOW	0.86/0.80	0.50/0.49	0.63/0.58	20,001
2	BOV	0.90/0.80	0.49/0.48	0.63/0.58	13,649
3	TF-IDF (Terms)	0.80/0.75	0.49/0.48	0.61/0.56	406,641
4	TF-IDF (Verbs)	0.89/0.81	0.52/0.49	**0.65/0.59**	46,373
5	LSA (20)	0.38/0.32	0.63/0.61	0.47/0.40	20
6	LSA (50)	0.52/0.46	0.64/0.62	0.58/0.50	50
7	LSA (100)	0.63/0.60	0.58/0.56	0.61/0.56	100
8	LSA (200)	0.73/0.75	0.55/0.53	0.62/0.59	200
9	LSA (400)	0.88/0.79	0.52/0.50	**0.65/0.59**	400
10	NMF (50)	0.26/0.35	0.50/0.52	0.34/0.33	50
11	NMF (100)	0.26/0.31	0.53/0.51	0.35/0.30	100
12	LDA (20)	0.27/0.23	0.56/0.57	0.37/0.31	20
13	LDA (50)	0.30/0.25	0.52/0.48	0.38/0.31	50
14	LDA (100)	0.38/0.35	0.58/0.56	0.46/0.41	100
15	LDA (200)	0.48/0.41	0.58/0.55	0.52/0.46	200
16	LDA (400)	0.51/0.43	0.59/0.55	0.55/0.45	400
17	HDP	0.45/0.41	0.51/0.50	0.48/0.43	150

metric which is a harmonic mean of precision and recall. We used 70% percent of the documents in our corpus for training the classifiers and 30% of the documents as a hold-out test set in a stratified fashion. We thus ensure the distribution of the target labels is roughly the same in our training and test sets.

5. Results and Discussion

Table 2 shows the results. We report precision, recall, and F1 scores both on the micro and macro level, but our main evaluation metric is macro F1.

TF-IDF (Verbs), the TF-IDF model that used n-grams of the verbs, and LSA (400) did best. On the positive side, this performance (and that of all the VSMs tested) is better than that of previously reported efforts (see Section 6).

On the other hand, the best-performing VSMs did only slightly better than the BOW or BOV models. These binary VSM models outperformed most of the VSM models with more complex, dimensionality-reducing representations. Since we expected that more complex VSMs might better reflect the legal factors and be better able to minimize feature engineering effort, this result was disappointing.

One way to explain these results is that our gold-standard corpus provides labels at the document level while trade secret factors are usually discussed on a sentence level, a problem also pointed out in [15]. Moreover, the labels are annotated mainly to study the interaction of factors in the trade secret domain, and there are some false negatives due to cases where the factor was mentioned but not applied in the decision (e.g., the factor may have been discussed in a description of a case cited for other reasons.) In an ideal scenario, the document representation should be able to filter-out noise and irrelevant case-specific information from the raw text files and aggregate information that discusses the factors actually applied and the issues actually decided. This may require the identification of sentence role types such as court's findings of fact. See [1, Chapter 11].

Table 3 shows the results of a best-performing model, TF-IDF (Verbs). Some factors may not have enough examples from which to learn. There also are excep-

Table 3. Results of a Best Model (TF-IDF Verbs)

Factor	P	R	F1	# ts	# tr	Top 5 Features (n-grams)
Security-Measures	0.76	0.9	0.83	29	62	hinged, submitted, think, snap, sold,
Info-Independently-Generated	0.89	0.53	0.67	15	43	tying, not_to_compete, snap, provided, find,
Disclosure-In-Negotiations	0.82	0.64	0.72	14	39	testified, not_to_compete, using, to_develop, taken,
Agreed-Not-To-Disclose	0.78	0.44	0.56	16	38	to_sell, sold, prevailing, hinged, design,
Brought-Tools	1	0.89	0.94	9	27	tying, disclosed, disclosing, hinged, affirm,
Restricted-Materials-Used	1	0.33	0.5	15	27	not_to_compete, erred, sold, shows, prevailing,
Info-Known-To-Competitors	1	0.44	0.62	18	24	shows, said, making, had, found,
Identical-Products	1	0.71	0.83	7	23	manufacturing, testified, developed, said, argues,
Disclosure-In-Public-Forum	1	0.33	0.5	12	21	said, contend, sitting save, sitting save read, given,
Unique-Product	1	0.3	0.46	10	21	found, not_to_compete, said, became, appropriated,
Secrets-Disclosed-Outsiders	1	0.38	0.55	8	17	manufacturing, denied, disclosed, were, claimed,
Outsider-Disclosures-Restricted	1	0.5	0.67	4	15	find, denied, sitting, submitted, desired,
No-Security-Measures	0	0	0	10	14	contained, said, think, found, selling,
Bribe-Employee	1	0.6	0.75	5	13	contend, enjoined, affirmed, implied, think,
Deception	0	0	0	3	13	argues, were, developed, acquired, sitting,
Agreement-Not-Specific	1	0.33	0.5	3	12	contained, not_to_compete, concluded, contend, conclude,
Vertical-Knowledge	1	0.6	0.75	5	11	said, think, existed, employed, continued,
Competitive-Advantage	0.75	0.6	0.67	5	10	erred, cited, tying, disclosing, affirmed,
Waiver-Of-Confidentiality	1	0.25	0.4	4	9	disclosed, found, denied, contained, said,
Employee-Sole-Developer	1	1	1	1	8	found, find, held, argues, affirmed,
Info-Reverse-Engineered	1	0.6	0.75	5	7	testified, not_to_compete, reverse, contend, using,
Noncompetition-Agreement	1	0.67	0.8	3	6	denied, tied, think, concerning, referred,
Info-Obtainable-Elsewhere	0	0	0	1	6	found, to_be, said, testified, using,
Invasive-Techniques	1	1	1	1	4	denied, were, developed, found, manufacturing,
Knew-Info-Confidential	1	0.75	0.86	4	4	were, concerning, erred, claimed, known,
Info-Reverse-Engineerable	0	0	0	4	1	testified, denied, found, contained, using,

tions like [Employee-Sole-Developer] or [Invasive-Techniques] that resulted in perfect classification despite the lack of training data and factors like [No-Security-Measures] that resulted in F1 score of 0.0 despite having 24 examples cases in the gold-standard corpus. We could explain this observation based on the fact that we have another factor [Security-Measures] which is closely related to the [No-Security-Measure] factor and might have caused some ambiguity for the classification. One could update the domain model based on this observation and merge these two factors into a single binary factor that takes values True or False.

The five most predictive features for each factor in Table 3 indicate that the SVM classifier has learned some promising features. [Info-Independently-Generated] and [Disclosure-In-Negotiations] each have the verb "not_to_compete" with F1-scores of 0.67 and 0.72 respectively. These results shows some potential for applying VSMs in the legal domain with minimal domain modeling.

6. Related Work

As noted, all of the VSM models outperformed the results reported by Ashley and Brüninghaus [4] with respect to the macro F1. In project SMILE, the researchers tested three representation schemes trying to predict the factors from the IBP corpus (of which VJAP's case base is a subset) [4] [14]. The first representation (BOW) was a bag-of-words representation similar to that of our bag-of-words VSM. In the second representation (RR), they replaced the parties and product names with their roles in the case. The third representation (ProPs) utilized the dependency parse results and converted each sentence within the case into (subject, verb), (verb, object), (verb, prepositional phrase), and (verb, adjective) tuples. They also performed additional processing to the negated verbs and passive verb forms within each sentence. However, the results were suboptimal (reported average F1=0.21) mainly due to the large dimensionality of the *bag-of-words* space and the lack of training data for each factor. Wyner and Peters [15] tried to solve this problem by starting from the description of the factors and using WordNet[6] expansions and expert knowledge to generate *factoroids*, plausibly semantic terms that are related to each factor. They used factoroids to generate rules as a part of GATE system[7] to annotate cases with respect to factors and pointed out the utility of creating a gold-standard corpus for machine learning.

In e-discovery, unsupervised learning enables exploratory clustering of documents and selecting seed sets for supervised learning. For example, the Categorix system clusters documents for review using PLSA, a probabilistic alternative to LSA as we used [6]. In earlier work, Uyttendaele, et al. applied an unsupervised, non-hierarchical clustering method and a TF-IDF vector space model like our TF-IDF (Terms) VSM to group paragraphs in court opinions thematically for purposes of summarization [7]. Schweighofer and Merkl applied self-organizing maps, a kind of unsupervised neural network, to explore and cluster documents in a corpus of European legal texts concerning public enterprises [8].

Lu, et al. clustered and segmented legal documents by topic in a huge corpus including judicial opinions and statutes [9]. The clustering process, however, used metadata unavailable to us including document citations, user behavior data, and topical classifications, which do not appear to capture topical information as detailed as trade secret factors. Winkels, et al. applied unsupervised learning to identify natural clusters of case citations of statutes, (as opposed to clusters of cases themselves as we do) for eventual use in a legal recommender context [10].

More recently, Panagis, et al. applied non-negative matrix factorization to a large set of judgments from the EU Court of Justice and European Court of Human Rights and selected clusters using topic coherence via word2vec to study topic drift over time [11]. Landthaler, et al. employed word embeddings (word2vec) in extracting similar obligations from the text of an EU Data Protection Directive 94/46/EC (EU-DPD) and similar provisions from a collection of German rental contracts [12]. We used NMF but did not employ word embeddings and leave it for future work as a potential substitute for WordNet expansion. Most recently, McCarty has called for an unsupervised approach to learning legal semantics in a corpus of unannotated cases to generate structured case notes [13].

7. Conclusion and Future Work

We used Vector Space Models to identify legal factors in in trade secret misappropriation cases. Factors, complex categories that capture a claim's substantive strengths and weaknesses, are intermediaries between statutory legal elements and cases' particular facts. Our results show that with simple heuristics and off-the-shelf components, one can detect some signal (i.e., features) for classifying

[6]https://wordnet.princeton.edu/
[7]https://gate.ac.uk/

factors in case texts. Our VSMs performed better than a previously published attempt at learning to identify factors in cases. On the other hand, our simplest VSMs outperformed most of the more complex ones, suggesting that dimensionality reduction did not add much if anything to classification performance.

We will study the *latent* dimensions learned by LSA, LDA, NMF, or HDP, to find mappings between what the model learns and legal factors. For example, one of the LSA model's latent dimensions contains the following verbs ordered by frequency: was used, was acquired, and to make, produce, obtain, manufacture, solicit, determine, establish, develop, show, prevent, design, gain, compete, treat, and enjoin. This latent dimension is related to the [Information-Used] branch of the domain model (Figure 1). One may learn a frequency, top-n threshold, or heuristic to identify factors with a finer granularity under this branch.

Such methods may pre-process case texts with factor-related information, so that human reviewers can confirm factor classifications more efficiently.

References

[1] Ashley, K. Artificial Intelligence and Legal Analytics. Cambridge University Press, 2017.
[2] Ashley K. Modeling Legal Argument: reasoning with cases and hypotheticals. The MIT Press, Cambridge, Massachusetts; 1990.
[3] Aleven V. Teaching case-based argumentation through a model and examples Ph.D. thesis, University of Pittsburgh, Pittsburgh, PA, USA.
[4] Ashley K, Brüninghaus S. Computer models for legal prediction. Jurimetrics 2006:309-352.
[5] Grabmair M. Modeling Purposive Legal Argumentation and Case Outcome Prediction using Argument Schemes in the Value Judgment Formalism: U. Pittsburgh; 2016.
[6] Privault C, ONeill J, Ciriza V, Renders J-M. A new tangible user interface for machine learning document review. Artificial Intelligence and Law 2010;18(4):459-479.
[7] Uyttendaele C, Moens M-F, Dumortier J. Salomon: automatic abstracting of legal cases for effective access to court decisions. Artificial Intelligence and Law 1998;6(1):59-79.
[8] Schweighofer E, Merkl D. A learning technique for legal document analysis. ICAIL 1999. ACM Press. p 156-163.
[9] Lu Q, Conrad J, Al-Kofahi K, Keenan W. Legal document clustering with built-in topic segmentation. 2011. 20th ACM Int'l Conf. Info. and Knowledge Management. p 383-392.
[10] Winkels R, Boer A, Vredebregt B, van SOMEREN A. Towards a Legal Recommender System. 2014. JURIX. 271:169-178.
[11] Panagis Y, Christensen ML, Sadl U. On Top of Topics: Leveraging Topic Modeling to Study the Dynamic Case-Law of International Courts. JURIX 2016. p 161-166.
[12] Landthaler J, Waltl B, Holl P, Matthes F. Extending Full Text Search for Legal Document Collections Using Word Embeddings. JURIX 2016. p 73-82.
[13] McCarty LT. Discussion Paper: On Semi-Supervised Learning of Legal Semantics. 2017.
[14] Ashley KD, Brüninghaus S. Automatically classifying case texts and predicting outcomes. Artificial Intelligence and Law 2009;17(2):125-165.
[15] Wyner AZ, Peters W. Lexical Semantics and Expert Legal Knowledge towards the Identification of Legal Case Factors. JURIX 2010. p 127-136.
[16] Fan RE, Chang KW, Hsieh CJ, Wang XR, Lin CJ. LIBLINEAR: A library for large linear classification. Journal of machine learning research. 2008;9(Aug):1871-4.
[17] Moschovakis YN. Sense and denotation as algorithm and value. Lecture notes in logic. 1994;2:210-49.
[18] Jurafsky D, Martin JH. Vector Semantics. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (3rd ed draft chapter 15-16). 2017.
[19] Wijaya DT. VerbKB: A Knowledge Base of Verbs for Natural Language Understanding (Doctoral dissertation, Carnegie Mellon University).