Improving Sentence Retrieval from Case Law for Statutory Interpretation

Jaromir Savelka
Intelligent Systems Program
University of Pittsburgh
jas438@pitt.edu

Huihui Xu
Intelligent Systems Program
University of Pittsburgh
huihui.xu@pitt.edu

Kevin D. Ashley
Intelligent Systems Program,
Learning Research and Development Center, School of Law
University of Pittsburgh
ashley@pitt.edu

ABSTRACT

Statutory texts employ vague terms that are difficult to understand. Here we study and evaluate methods for retrieving useful sentences from court opinions that elaborate on the meaning of a vague statutory term. Retrieving sentences instead of whole cases may spare a user the need to review long lists of cases in search of useful explanations. We assembled a data set of 4,635 sentences that were responses to three statutory queries and labeled them in terms of their usefulness for interpretation. We have run a series of experiments on this data set, which we have made public, assessing different techniques to solve the task. These include techniques that measure the similarity between the sentence and the query, utilize the context of a sentence, expand queries, or assess the novelty of a sentence with respect to a statutory provision from which the interpreted term comes. Based on a detailed error analysis we propose a specialized sentence retrieval framework that mitigates the challenges of retrieving case law sentences for interpreting statutory terms. The results of evaluating different implementations of the framework are promising (.725 for NDGC at 10, .662 at 100).

CCS CONCEPTS

• **Information systems** → **Information retrieval**; *Retrieval models and ranking*; Rank aggregation; Similarity measures.

KEYWORDS

Information retrieval, statutory interpretation, case-law analysis, relevant sentences

ACM Reference Format:
Jaromir Savelka, Huihui Xu, and Kevin D. Ashley. 2019. Improving Sentence Retrieval from Case Law for Statutory Interpretation. In *Seventeenth International Conference on Artificial Intelligence and Law (ICAIL '19), June 17–21, 2019, Montreal, QC, Canada,* Floris Bex (Ed.). ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3322640.3326736

ICAIL '19, June 17–21, 2019, Montreal, QC, Canada

ACM ISBN 978-1-4503-6754-7/19/06…$15.00
https://doi.org/10.1145/3322640.3326736

1 BACKGROUND AND MOTIVATION

Statutes are written laws enacted by legislative bodies. They set forth the collection of legal norms which are legally binding rules of conduct. A single statute is usually concerned with a specific area and comprises provisions that express the individual legal rules (e.g., rights, prohibitions, duties). Understanding statutory provisions is difficult because the abstract rules they express must account for diverse situations, even those not yet encountered. Provisions of law communicate general standards and refer to classes of persons, acts, things, and circumstances. [16, p. 124] Therefore, legislators must use vague [9], open textured [16] terms, abstract standards [10], principles, and values [5] to deal with this uncertainty.

For example, consider the two emphasized phrases from the following (abridged) example provision (29 U.S. Code § 203):

> "Enterprise" means the *related activities* performed [...] for a *common business purpose* [...].

Understanding of the provision depends on a well-developed knowledge of the meaning of the two emphasized phrases. Doubts about the meaning of a provision may be removed by interpretation. [23]

The meaning of the phrase *common business purpose* from the example provision could play a pivotal role in case of, for instance, determining if two restaurants in different parts of the same city, sharing a single owner, constitute an "enterprise" within the meaning of the provision. The interpretation involves an investigation of how the term has been referred to, explained, interpreted or applied in the past. This is an important step that enables a user to then construct arguments in support of or against particular interpretations.

Searching through a database of statutes, court decisions, or law review articles, one may stumble upon sentences such as these:

(1) Courts have held that a joint profit motive is insufficient to support a finding of *common business purpose.*
(2) The fact of common ownership of the two businesses clearly is not sufficient to establish a *common business purpose.*
(3) Because the activities of the two businesses are not related and there is no *common business purpose*, the question of common control is not determinative.
(4) The problems then are whether we have related activities and a *common business purpose.*
(5) The third test is "*common business purpose.*"

Some of these sentences are useful for interpreting the phrase (1 and 2). Some of them look like they may be useful (3) but the rest appears to have very little (4) if any (5) value. Reviewing such sentences manually is labor intensive due to high redundancy and the large number of sentences that simply quote the statutory language.

Our goal is to evaluate and propose computational methods to support this task. Specifically, given a user's interest in the meaning of a particular statutory phrase, we would like to rank more highly those sentences the goal or effect of which is to elaborate upon the meaning of the statutory phrase of interest, such as:

- definitional sentences (e.g., a sentence that provides a test for when the phrase applies)
- sentences that state explicitly in a different way what the statutory phrase means or state what it does not mean
- sentences that provide an example, instance, or counterexample of the phrase
- sentences that show how a court determines whether something is such an example, instance, or counterexample.

By contrast, sentences that merely quote or closely paraphrase the statutory language should be demoted. Even though they may contain instances of the statutory phrase such as "related activities" or "common business purpose," they do not help because they contain no additional information about the meaning of the phrase.

2 RELATED WORK

In [41] the researchers annotated text passages with definitions from the non-fact-reporting sections of forty cases as Core, Addition, or Definiendum. Core sentences are self-contained and capture an essential feature of the defined concepts. Addition sentences provide elaborations, details, or clarifications. Definiendum sentences merely mention a term being defined to which a subsequent Core or Addition sentence refers anaphorically. The researchers identified a set of patterns (templates) of indicators for a subset of the definitions. They then operationalized these patterns with rule-like expressions. [41] Given the complexity of legal texts, the limitations of parsing, and the difficulty of manually constructing the pattern-matching expressions, they explored using ML to generate and test new expressions, but with limited success.

Subsequent work focused on extracting definitions from the German Civil Code using rule based annotation [7, 17, 42] and ML [6]. In [14] the authors used supervised learning to classify sentences from landlord/tenant provisions and rental agreements by functional type, including definitions, whose "primary function ... is to describe and clarify the meaning of a term within the law." F(1) for definitions was .87.

Recent work on definitional question-answering from domain-specific but less well-structured texts is instructive. Question-answering systems have been applied in specialized domains like healthcare to answer terminological questions such as, "What is irritable bowel syndrome?" or "What does TB mean?" Given a question, QA systems (a) determine the nature of the question and the kind of answer anticipated, (b) retrieve and select sentences relevant to the definiendum and, (c) given the kind of answer, "choose non-redundant definition sentences from the overall results". [30]

Sentence retrieval is more challenging than ordinary document retrieval: "(1) the brevity of sentences vs. documents exacerbates the usual term-mismatch problems, and (2) the verbosity of questions can lead to critical query terms being obscured by supporting terms." [28] TF-ISF is a successful term-weighting strategy in which a sentence's relevance to a query increases with the number of times the query term appears in the sentence [11, 28].

The authors of [11] take into account the local context of a candidate sentence in the document in which it is found, that is, the sentences that surround it in the document and the sentence's importance within the document. Their model can estimate "how well the sentence explains both the document and the query topic" and "include a query-independent probability that encodes the importance of a sentence in a document." They demonstrate empirically their model's contribution to retrieval efficacy. See also [8] for improved ways to account for local context.

Using definitional soft pattern matching, a kind of instance-based learning, and information gleaned from definitional websites, [4] demonstrated that unsupervised learning of soft matching patterns performed better than rule induction or manually-constructed rules, especially when integrating the web information for adjusting ranking weights. By extracting evidence from web knowledge base articles about the definiendum [4], the weight of terms appearing within the articles is augmented [12]. Another approach focuses on obtaining patterns "across several definitions of the same context or type (e.g., 'is → novelist → romantic')" [12].

The focus on context models begs the question of which types of definitional patterns the legal domain presents and how can they be automatically extracted from legal case texts. As [41] observed, there are other ways of conveying the meaning of a statutory phrase beside definitions. [41] identified paratactic definitions in which "the definiens is somewhere in the environment of the definiendum, but is not related to it through a common clause level predicate." As noted, in interpreting a statutory phrase, a court may also provide a test for when the phrase applies, describe what the phrase means or does not mean, or provide an example, instance, or counterexample of the phrase. These approaches to specifying the meaning of a statutory phrase all appear in legal argument, and, presumably each is associated with various sentential patterns.

3 DATA SET

We downloaded the complete bulk data from the Caselaw access project.[1] This includes all official, book-published U. S. cases from all federal and state courts as well as from a number of territorial courts [33]. Altogether the data set comprises more than 6.7 million unique cases. More detailed statistics are reported in Table 1 in the All row. We ingested the data set into an Elasticsearch instance.[2] For the analysis of the textual fields we used the LemmaGen Analysis plugin[3] which is a wrapper around a Java implementation[4] of the LemmaGen project.[5] The lemmatizer is based on so-called induced ripple-down rules. [20]

To support our experiments we indexed the documents at multiple levels of granularity. Specifically, the documents were indexed at the level of full cases as well as segmented into the head matter and individual opinions (e.g., majority opinion, dissent, concurrence). This segmentation was performed by the Caselaw access project using a combination of human labor and automatic tools.[6] We also

[1] A small portion of the data set is available at case.law. The complete data set could be obtained upon entering into research agreement with LexisNexis.
[2] https://www.elastic.co/
[3] https://github.com/vhyza/elasticsearch-analysis-lemmagen
[4] https://github.com/hlavki/jlemmagen
[5] http://lemmatise.ijs.si
[6] Per information provided in the email from info@case.law on 2019-01-07.

Table 1: Data set summary. IEV stands for the 'independent economic value', IP for 'identifying particular', and CBP for 'common business purpose'.

	cases	opinions	paragraphs	sentences
IEV	872	888	1435	1538
IP	1882	1892	2134	2217
CBP	371	377	735	880
Total	3125	3157	4304	4635
All	6715418	13958364	206058346	538680570

used the U.S. case law sentence segmenter from [39] to segment each case into individual sentences and indexed those as well. Finally, we used the sentences to create paragraphs. We considered a line-break between two sentences as an indication of a paragraph boundary. Including indexes the resulting data set is 622GiB in size and has nearly 0.8 billion documents.

We queried the system for sentences mentioning three terms from different provisions of the U. S. Code (the official collection of federal statutes).[7] As in [35] the terms were 'independent economic value' (IEV) from 18 U.S. Code §1839(3)(B), an 'identifying particular' (IP) from 5 U.S. Code §552a(a)(4), and 'common business purpose' (CBP) from 29 U.S. Code §203(r)(1). The terms were selected on the basis of being vague and coming from different areas of regulation. We did not include more terms due to the high cost of labeling. However, we have acquired resources for future extension.

We did not limit ourselves to sentences coming from the 20 top decisions for each term as in [35]. This led to a significantly larger data set of 4,635 sentences as opposed to 243. The detailed statistics are reported in Table 1. As in [35], the authors here classified the sentences in terms of four categories with respect to their usefulness for the interpretation of the corresponding statutory term:

(1) **High value** – This category is reserved for sentences the goal of which is to elaborate on the meaning of the term.
(2) **Certain value** – Sentences that provide grounds to draw some conclusions about the meaning of the term.
(3) **Potential value** – Sentences that provide additional information beyond what is known from the provision.
(4) **No value** – Sentences that do not provide any additional information over what is known from the provision.[8]

As reported in [35], annotators need to be properly trained for this challenging task. We adopted multiple measures to ensure the annotations of the resulting data set are of high-quality. The final labels were either assigned by consensus of all three annotators or on the basis of the third annotator's independently resolving differences between the other two annotators' sentence labels. For each term we selected a subset of sentences that were labeled first. Once these subsets were finished the annotators met to resolve any systematic disagreements. Only after that meeting did we proceed to label the rest of the sentences.

To measure inter-annotator agreement we used Krippendorff's α [21] which is designed for situations of more than 2 annotators, any type of labels (including ordinal as in our case), and missing

[7] https://www.law.cornell.edu/uscode/text/
[8] The annotation guide is at https://github.com/jsavelka/statutory_interpretation.

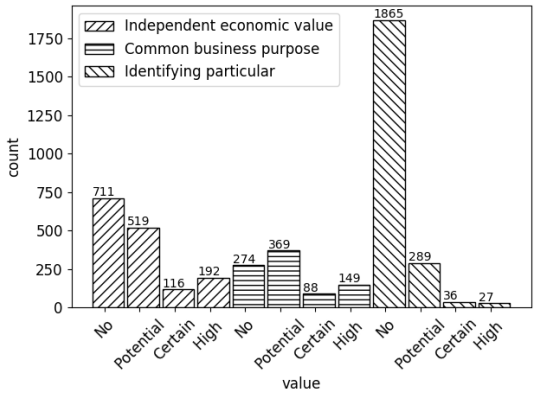

Figure 1: Sentence value distribution

data (i.e., each data point was independently labeled by 2 of the 3 annotators). We confirmed that the task is challenging and requires annotator training. After the first labeling round on IEV the α was only 0.30. After the resolution of the systematic differences, however, the α rose to 0.55 on the whole IEV subset. Because of the strategy for resolving systematic disagreements early in the annotation process, we achieved higher agreement than in [35] —α for IP is 0.82 and 0.78 for CBP. The overall α of 0.79 is typically interpreted as substantial agreement.

Figure 1 reports counts for the final consensus labels. The most frequent is the No value label for IEV and IP and the Potential value label for CBP. The more valuable labels (Certain and High) are less frequent. This is especially true for IP where these labels are extremely rare. To support research in sentence retrieval for statutory interpretation we release the resulting data set of 4,635 labeled sentences to the public.[9]

4 EXPERIMENTS

It appears to be well-established that the traditional approach to ranking, based on computing similarity between the query and retrieved documents, is less effective for very short documents such as sentences (see, e.g., [29]). Usually the main cause of the decreased performance is the lack of a robust overlap between a query and a sentence. The similarity between the query and a document is based either on a direct overlap of terms or on an "overlap" in their meanings. In larger documents an unusually high occurrence of a term or its meaning strongly indicates that the document might be "about" the term. This "aboutness" assumption often works surprisingly well for longer documents. In shorter documents there is typically just one exact occurrence of the term. Even in case of more than one occurrence it is questionable if the "aboutness" assumption would still be as solid as for longer documents. As a result, there are no grounds on which the short documents could be ranked reliably based on the term occurrence statistics.

[9] https://github.com/jsavelka/statutory_interpretation

4.1 Retrieving Sentences Directly

We investigate the effectiveness of the direct approach to ranking, based on computing similarity between the query (the term of interest) and retrieved documents (sentences mentioning the term of interest), in the context of retrieving case law sentences for statutory interpretation. First, we retrieve all sentences that contain the term of interest. Using different strategies for measuring similarity of a sentence and a query, sentences are then ranked from the most similar to the least. The motivation is that the more similar the sentence is to the term of interest, the more likely it is about the term, and hence it may be useful for explaining its meaning.

We experiment with different strategies of measuring similarity to analyze what impact this has on the performance. Merely as a reference point we report the performance of a random system on a large sample of repeated runs. After each run we compute the average performance over all the preceding runs. We stop the iteration once the average remains stable for 10,000 consecutive runs. The performance of the random baseline is then the average performance to which the system converged.

We first evaluate an approach to measuring similarity of query and document based on the Okapi BM25 function, from the well know TF-IDF family. The function is defined as follows:

$$\text{BM25} = \sum_{t \in q} TF \cdot IDF \cdot QTF$$

$$\text{TF} = \frac{(k_1 + 1) \cdot tf_{td}}{k_1 \cdot \left(1 - b + b \cdot \frac{L_d}{L_{avg}}\right) + tf_{td}}$$

$$\text{IDF} = log\left(\frac{N - df_t + \frac{1}{2}}{df_t + \frac{1}{2}}\right) \quad \text{QTF} = \frac{(k_3 + 1) \cdot tf_{tq}}{k_3 + tf_{tq}}$$

Here, N is the size of the collection, df_t is the number of documents in which t occurs, tf_{td} is the frequency of term t in document d, tf_{tq} is the frequency of term t in query q, L_d and L_{avg} are the length of d and the average document length for the whole collection. k_1, k_3 and b are tuning parameters. We use the typical values of $k_1, k_3 = 1.2$ and $b = 0.75$. [24, p. 233]

The second approach is a variant of TF-IDF tailored to accommodate some specifics of short documents such as sentences. The measure was presented in [1] and was subsequently referred to as TF-ISF [8, 11, 28]:

$$\text{TF-ISF} = \sum_{t \in q} log(tf_{td} + 1) \cdot log\left(\frac{N + 1}{\frac{1}{2} + df_t}\right) \cdot log(tf_{tq} + 1)$$

The three components of the formula are adapted versions of TF, IDF, and QTF. The meaning of the notation is the same as in the previous formula.

The language modeling approach to IR implements the idea that a document is a good match to a query if its language model is likely to generate the query. We work with a simple query likelihood language model presented in [32]:

$$P(d|q) \propto P(d) \prod_{t \in q} ((1 - \lambda)P(t|M_c) + \lambda P(t|M_d))$$

Here, M_d is the language model of document d, M_c is a language model built from the entire document collection, and $0 < \lambda < 1$ is a hyperparameter. Correctly setting λ is important to the performance of this model. The equation captures the probability that the document that the user had in mind was d. [24, pp. 237–252] We optimized λ on IEV@100 where the best value is around 0.9.

Finally, we measure the similarity between projections of a query and a document in a low dimensional semantic space. In order to achieve this, we work with vector representations of words referred to as word embeddings. These representations are motivated by the so-called distributional hypothesis claiming that words that are used and occur in the same contexts tend to purport similar meanings [13, 15]. Regardless of a method by which word embeddings are generated, the idea is that the words with similar meanings are projected onto the vectors the cosine similarity of which is high. We use Gensim [34] to work with embeddings trained with various algorithms on a number of different corpora.

From the numerous available options we chose to experiment with word embeddings generated by the popular word2vec algorithm based on the skip-gram model [25, 26], the GloVe model that combines global matrix factorization and local context window methods [31], and the FastText algorithm based on the skip-gram model, where each word is represented as a bag of character n-grams [2, 18, 19]. Furthermore, we use all the 538,680,570 sentences from our corpus (Section 3) to train domain specific embeddings (w2vec*). To train the embeddings we used the sequences of lemmatized tokens in lower case as described in Section 3. Before training the embeddings we trained a model for detection of 2- and 3-word long phrases to be included in the embeddings' training.[10] [3, 26] We trained embeddings with 300 dimensions and finished the training after 5 iterations over the corpus.[11]

The similarity is determined by (1) projecting the query $\mathbf{q} = \{t_1, t_2, \dots\}$ and document $\mathbf{d} = \{t_1, t_2, \dots\}$ onto the vectors $\vec{q}$ and $\vec{d}$ that are from the same low dimensional semantic space, and (2) measuring the distance between $\vec{q}$ and $\vec{d}$ using the cosine of the angle between the two vectors:

$$\text{COS} = \frac{\sum_{i=1}^{n} \vec{q}_i \cdot \vec{d}_i}{\sqrt{\sum_{i=1}^{n} \vec{q}_i^2} \cdot \sqrt{\sum_{i=1}^{n} \vec{d}_i^2}}$$

Here, $\vec{q}$ is an averaged normalized vector over $\vec{t_i}$ in query $\mathbf{q}$, $\vec{d}$ is an averaged normalized vector over $\vec{t_i}$ in document $\mathbf{d}$, and n is the dimensionality of the vectors. We use the word2vec model (300 dimensions) trained on the Google News corpus (about 100 billion words)[12] [25–27], the GloVe model (300 dimensions) trained on the Wikipedia 2014 and Gigaword 5 corpora (6 billion tokens)[13] [31],[14] FastText model (300 dimensions) trained on Wikipedia,[15] [2] and the model trained on our corpus described above (w2vec*).

[10] We used Gensim's phrases module which is available at https://radimrehurek.com/gensim/models/phrases.html
[11] See https://radimrehurek.com/gensim/models/word2vec.html.
[12] https://code.google.com/archive/p/word2vec/
[13] https://nlp.stanford.edu/projects/glove/
[14] Both available at https://github.com/RaRe-Technologies/gensim-data
[15] https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

4.2 Smoothing Sentences with Context

We analyze the effects of considering the context of a sentence when deciding about its ranking. As in case of the previous batch of experiments (Subsection 4.1), we first retrieve all the sentences that contain the term of interest. Unlike in the first batch we also retrieve the remaining parts of the decisions from which the sentences come. Using the same similarity measuring strategies as before we ranked the sentences based on their similarity to the query as well as the similarity of their varying contexts to the query. The motivation behind this approach is that parts of a decision other than the sentence itself could be suggestive about the value of the sentence. For example, if we determine that the whole decision is "about" the term of interest then we expect that at least some of the sentences it contains will be "about" the term—and these are the sentences that we would like to rank highly.

The first approach we experiment with to incorporate context into the similarity measurement is linear interpolation applied to similarities as measured on the sentence itself and on a fixed context. The types of context are the whole case, an opinion, and a paragraph (as described in Section 3). We introduce the hyperparameter λ_1 that controls the weighting of the two pieces (the sentence itself and the context). The general form of this approach is then:

$$Sim\text{-}i = (1 - \lambda_1)Sim_s + \lambda_1 Sim_i$$

Here, Sim_s is the similarity as measured between the sentence and the query, Sim_i is the similarity as measured between the context and the query (where $i \in$ {c(ase), o(pinion), p(aragraph)}), and λ_1 controls the weight assigned to each component (the higher λ_1 the more importance the context is given).

The general form has slightly varying implementations across the tested similarity functions. For both BM25 and TF-ISF, the implementation is as follows:

$$\text{BM25-i, TF-ISF-i} = \sum_{t \in q} [(1 - \lambda_1)TF_s \cdot IDF_s + \lambda_1 TF_i \cdot IDF_i] \cdot QTF$$

The QLLM already had one hyperparameter (λ) in its original form. Thus, the context-aware form needs to accommodate two hyperparameters where λ_2 stands for the original λ:

$$P(d|q) \propto P(d) \prod_{t \in q} [(1 - \lambda_1 - \lambda_2)P(t|M_c) + \lambda_1 P(t|M_i) + \lambda_2 P(t|M_d)]$$

The new M_i element is the language model of context i. Finally, the interpolated cosine similarity for representations based on word embeddings is straightforward:

$$\text{COS-i} = (1 - \lambda_1)COS_s + \lambda_1 COS_i$$

The second approach we evaluate is the recursive interpolation presented in [8] originally designed for TF-ISF. The idea is to consider n preceding and n following sentences as the context of the sentence we wish to evaluate. The sentences that are closer to the focused sentence are assigned more weight than those that are further away. The general formula for this approach is the following:

$$Sim\text{-}r_s = (1 - \lambda_1)Sim_s + \lambda_1[Sim\text{-}r_{s-} + Sim\text{-}r_{s+}]$$

Here, Sim_s is the similarity score computed between the focused sentence and the query, $s-$ and $s+$ stand for the preceding and the next sentence respectively, and λ_1 is the hyperparameter controlling the weight assigned to the context. The authors of [8] set n to 3 suggesting that it could be extended to the whole document if necessary. However, we found that for $n > 4$ the computation becomes expensive. Since we observed increased performance with larger n we set n to 4. The implementation for the specific methods is straightforward where Sim_s is replaced with the respective method.

4.3 Expanding Query with the Provision

We assess the effects of extending the query with (1) other words from the source provision on top of those that are part of the term of interest, and (2) the top 500 most similar words produced by the word2vec embeddings trained on our corpus. Using various similarity measuring strategies we ranked the sentences based on their similarity to the query as well as their similarity to the extended query. The motivating factor was the assumption that sentences that are "about" the term of interest from the provision are likely to contain other words from the provision or the words that are closely related to the term of interest.

The assumption turned out to be wrong in the context of the task of retrieving case-law sentences for statutory interpretation. As in the experiments on smoothing sentences with context we used linear interpolation to control how a model is informed by the original query versus the expanded query. Optimizing the hyperparameter λ_1 with respect to IEV the models always settled on ignoring the expanded query altogether. This made the models equal to those that retrieve the sentences directly (Subsection 4.1).

Because of this unfortunate development we steered the experiment in a different direction. Instead of comparing the similarities of the query and the expanded query to the sentences themselves, we compared them to the whole cases. In addition, we forced the system to use the expanded query and ignore the original one. The intuition behind this approach was that, given we only work with sentences that mention the term of interest, the cases that are most similar to the expanded queries either are focused on:

(1) the source provision (in case the query is expanded with the words from the source provision)
(2) the term of interest (in case the query is expanded with the top 500 similar words)

Since these types of cases are highly likely to contain the sentences we would like to retrieve, we assumed that the models could be useful either on their own or as a part of some variant of the compound system described in Subsection 4.5.

The specific implementations of the models measuring the similarities between expanded queries and whole cases are straightforward. The formulas presented in Subsection 4.1 (direct retrieval) apply verbatim—except the query is replaced with the expanded query and sentences are replaced with whole cases.

4.4 Novelty Detection

In this set of experiments we investigate the effectiveness of focusing on the amount of information a sentence provides over what is known from the provision. As in the other sets of experiments

we first retrieve all the sentences that contain the term of interest. We experiment with strategies to measure how much content the sentence adds with respect to the provision, as well as with measuring a distance between the two documents. Our relevance definition motivates the focus on novelty; sentences that do not provide additional information are deemed as having no value.

As the first approach to measure a sentence's additional content, we evaluate the number of new words it includes, i.e., the number of words that appear in the sentence that are not contained in the provision. This simple measure is often surprisingly effective. [1]

$$NW = \left|\{w_i | w_i \in \mathbf{S}_j \setminus \{w_i | w_i \in \mathbf{P}\}\right|$$

$\mathbf{S}_j$ is the set of words w_i from a sentence and $\mathbf{P}$ is the set of words w_i from the source provision.

The second approach is closely related to the first one. The only difference is that we control for the size of the sentence. Thus, instead of the number of the new words we work with a ratio of the new words within a sentence:

$$NWR = \frac{NW}{\left|\{w_i | w_i \in \mathbf{S}_j\right|}$$

In addition we measure Word Mover's Distance (WMD) on the embedding representations of a sentence and the source provision. WMD captures the dissimilarity between two documents $\mathbf{d}$ and $\mathbf{d}'$ as the minimum distance that the embedded words of one document need to "travel" to reach the embedded words of another document. The distance is computed by solving the following linear program:

$$\begin{aligned} &\min_{\mathbf{T} \geq 0} \sum_{i,j=1}^{n} \mathbf{T}_{ij} c(i,j) \\ &\text{subject to:} \sum_{j=1}^{n} \mathbf{T}_{ij} = d_i \qquad \forall i \in \{1, \dots n\} \\ &\qquad\qquad \sum_{i=1}^{n} \mathbf{T}_{ij} = d'_i \qquad \forall j \in \{1, \dots n\} \end{aligned}$$

Here, $\mathbf{d}$ and $\mathbf{d}'$ are the bag of word vector representations of two documents (a sentence and the source provision in our case), $\mathbf{T} \in \mathbb{R}^{n \times n}$ is a flow matrix where $\mathbf{T}_{ij} \geq 0$ denotes how much of word i in $\mathbf{d}$ travels to word j in $\mathbf{d}'$. [22]

4.5 Compound Models

In the final batch of experiments we explore the effects of combining models evaluated previously. For each sentence we compute scores based on the most successful models that (1) utilize the sentence's context, (2) work with an expanded query, and (3) measure the sentence's novelty with respect to the source provision. The motivation is to inform the relatively successful context-aware models with different types of signals coming from the models based on query expansion and novelty detection. Based on the error analysis of the results on IEV (Section 6) we propose a task specific sentence retrieval model that has the following general form:

$$\text{CMP-rank}(q, s, sp, \mathbf{C}) = \textit{Sim-i}(q, s, \mathbf{C}_i) \cdot \textit{NI}(s, sp) \cdot \textit{DDI}(q, \mathbf{C}_j)$$

Here, q is the query, s is a sentence, sp is the source provision, $\mathbf{C}$ is the set of available contexts ($\mathbf{C}_i$ is a specific context where $i \in \{c, o, p, r\}$), *Sim-i* is the base ranking function from the family of the context-aware models, *NI* is the novelty indicator function from the family of the novelty detection models, and *DDI* is the different domain indicator function from the group of the models utilizing query expansion. *NI* mitigates the problem of retrieving sentences that are not much more than partial or complete citations of the source provision. *DDI* tackles the problem of retrieving cases that discuss the term that is the same as the term of interest, yet it comes from a different domain (see Section 6 for details).

As concrete implementations of *Sim-i* on which to experiment we choose TF-ISF-p, BM25-p, QLLM-c, and w2vec-r. Apart from being the most successful models, these cover the full spectrum of similarity measuring techniques as well as the whole range of local and large contexts.

We utilize NewWrdRatio (NWR) for the *NI* component. This gives us the following implementation of *NI*:

$$\textit{NI}(s, sp) = \begin{cases} 1 & \text{if } \textit{NWR}(s, sp) \geq \lambda_1 \\ 0 & \text{if } \textit{NWR}(s, sp) < \lambda_1 \end{cases}$$

This component requires a sentence to surpass a set novelty threshold as measured by *NWR*. The threshold is controlled through hyperparameter λ_1. If the sentence fails to surpass the threshold it is effectively discarded by being assigned a score of 0. Even though NewWords was evaluated as slightly more successful than *NWR*, we have chosen to work with the latter because of its more intuitive interpretation. Knowing that a sentence has $NWR = 0.4$ is informative on its own whereas knowing that a sentence has $NW = 5$ is almost meaningless without further information about the sentence (at least its length). We heuristically set λ_1 to 0.2 by inspecting the output of *NWR* applied to the sentences from IEV. This corresponds to the lowest score of a sentence with high value.

We use TF-ISF-g as the *DDI* component. This yields the following implementation of *DDI*:

$$\textit{DDI}(s, c) = \begin{cases} 1 & \text{if TF-ISF-g} \geq \lambda_2 Avg_{10} \\ 0 & \text{if TF-ISF-g} < \lambda_2 Avg_{10} \end{cases}$$

Here, Avg_{10} is the average of the scores of the top 10% documents. This approximates the score of a document that comes from the same domain as the term we are interested in. The reason we do not take the score of the top document is to account for the possibilities of a case citing the source provision multiple times or of a short case the majority of which is the citation of the source provision. These would give an unrealistically high estimate for our purposes. The hyperparameter λ_2 then controls the threshold below which we consider documents to come from a different domain. Based on the error analysis of IEV we set $\lambda_2 = 0.5$ for our experiments.

4.6 Evaluation Metrics

Since our notion of relevance is non-binary we use normalized discounted cumulative gain (*NDCG*) to evaluate the performance of different approaches. An output of the presented ranking algorithms for each query q_j has the form of an ordered tuple of sentences $S_j = (s_1, s_2, \dots, s_n)$. We chose to evaluate the rankings at $k = 10$ and 100 which means that the tuples produced by the algorithms are

truncated to the respective lengths. For each query q_j the $NDGC$ at each k is then computed as:

$$NDGC(S_j, k) = \frac{1}{Z_{jk}} \sum_{i=1}^{k} \frac{rel(s_i)}{log_2(i+1)}$$

The function $rel(s_i)$ takes a sentence as an input and outputs its value in a numerical form. It is defined as follows:

$$rel(s_i) = \begin{cases} 3 & \text{if } s_i \text{ has high value} \\ 2 & \text{if } s_i \text{ has certain value} \\ 1 & \text{if } s_i \text{ has potential value} \\ 0 & \text{if } s_i \text{ has no value} \end{cases}$$

Z_{jk} is a normalizing quantity which is equal to $NDGC(S_j, k)$ where S_j is the ideal ranking. In our case this means that all the s_i with high value labels are at the beginning positions of the tuple, followed by those with the certain value, then potential value, and finally no value sentences.

The macro average over the set of queries Q is then computed simply as:

$$NDGC(\mathbf{S}, k) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} NDGC(S_j, k)$$

5 RESULTS

The results of the experiments described in Section 4 are summarized in Table 2. The first section of the table reports the results of the experiments on retrieving sentences directly (Subsection 4.1). The results indicate that the direct approach is not effective. Of all the six tested methods only TF-ISF outperformed the Random baseline by larger than a negligible margin. We further analyze this outcome in Section 6.

The second section of Table 2 reports the results of the experiments on smoothing sentences with context (Subsection 4.2)–'-c' for a whole case, '-o' for an opinion, '-p' for a paragraph and '-r' for the recursive context as described in [8]. The results suggest that taking advantage of a sentence's context improves the performance. Regardless of the approach, the type of context, or the similarity measuring method, the overall performance was almost always better than if the context was not taken into account. The best version of any of the methods was evaluated over 0.4 for both NDGC@10 and @100. None of the methods achieved this performance when applied in the absence of the sentence's context. The best method overall is the TF-ISF with a paragraph as the context.

The results from the experiments on query expansion (Subsection 4.3) are reported in the third section of Table 2. As mentioned earlier the experiments on query expansion that we originally had in mind, i.e., matching the query and the expanded query to the sentences themselves, did not pan out well. All the models optimized in such a way as to ignore the expanded query, making themselves equivalent to those reported in the first section of the Table. Therefore, we omit these in the query expansion section of the table. Instead, we list the performance of the models that match the expanded query—either with the words from the source provision ('-g' suffix) or with the top 500 similar words from the trained w2vec* model ('-e' suffix). Clearly the models based on matching the query expanded with the words from the source provision have better performance than the models based on matching the query expanded with the similar words. In any case, both types of models underperform their counterparts where the original (not expanded) query is matched to the whole case (the models with '-c' suffix in second section of Table 2). The reasons we believe these models to be useful are discussed later in Section 6.

The Novelty section of Table 2 reports the results of the experiments focused on measuring the amount of information a sentence adds to what is already known from the provision. It appears that the models measuring the extra amount of words (count and ratio) perform better than WMD based models that measure the semantic distance between a sentence and the provision. The best method overall is the number of new words closely followed by their ratio.

The results from the experiments on compound models (Subsection 4.5) are reported in the final section of Table 2. The first element in a model's name stands for the base ranker (Tp is TF-ISF-p, Bp is BM25-p, Qc is QLLM-c, and Wr is w2vec-r). The subsequent elements inform about the indicator functions applied to that model (Tg is TF-ISF-g, and Nr is NewWrdRatio). We observe that in most of the cases the application of the indicator components improve the performance of the base ranker. While the Tp-Tg-Nr is the most successful model overall, it appears that it was QLLM-c that benefited the most from the augmentation (Qc-Tg-Nr). Table 3 lists the top 5 results for each query generated by the Tp-Tg-Nr method.

6 DISCUSSION

We confirmed that the direct approach to sentence retrieval does not work well. In the context of our specific task the root cause of the low performance appears to be the preference of the systems for short over long sentences. For example, the following sentences top the rankings produced by almost all the systems:

(1) The "Independent Economic Value" Test
(2) It was Altavion's burden to show independent economic value.
(3) But again, the Producers' evidence of independent economic value was more theoretical than real.
(4) As the formula for an unreleased product, it has independent economic value.

From these sentences only (4) has High value whereas the rest have No or Potential value. Since the provision of additional valuable information is part of the relevance definition in our task the preference for short sentences is not a good strategy. That is the reason why most of the systems barely outperform the Random baseline.

The relative success of the TF-ISF method could be explained by the deliberate omission of the normalization based on a document length in the TF part. Unlike other studied methods, TF-ISF prefers longer sentences that mention the terms from the query multiple times over the shorter sentences presented above. This strategy is clearly preferable in sentence retrieval for statutory interpretation.

Our experiments clearly suggested that taking sentences' context into account improves the performance. The influence of the context over the ranking decision was controlled by the hyperparameter λ_1, where $\lambda_1 = 1$ means that the ranking decision is based entirely on the sentence in its context whereas $\lambda_1 = 0$ means that the decision is based solely on the sentence itself. Table 4 reports the values of λ_1 as optimized on IEV. From these values it is clear that the sentence in its context had much more influence than the sentence itself.

Table 2: Results of the experiments described in Section 4–Direct (4.1), Smoothing with context (4.2), Query expansion (4.3), Novelty (4.4), and Compound (4.5). The context suffixes are -c(ase), -o(pinion), -p(aragraph), and -r(ecursive). Query expansion suffixes are (ori)-g(inal), and -e(mbedding similar words). In the Novelty section the suffix is -w(md). In the Compound section there is T(F-ISF-)p, B(M25-)p, Q(MML-)c, W(2vec-)r, T(F-ISF-)g, and N(ewWrd)R(atio).

		IEV		IP		CBP		Overall			IEV		IP		CBP		Overall	
	Method	@10	@100	@10	@100	@10	@100	@10	@100	Method	@10	@100	@10	@100	@10	@100	@10	@100
Direct	Random	.288	.287	.066	.093	.375	.375	.243	.252	GloVe	.021	.187	**.284**	.288	.000	.322	.102	.266
	BM25	.042	.231	.220	.088	.000	.280	.087	.200	w2vec	.000	.151	.240	.375	.000	.326	.080	.284
	TF-ISF	**.617**	.245	.257	.325	**.874**	**.534**	**.583**	**.368**	fastt	.000	.120	.234	**.327**	.000	.309	.078	.252
	QLLM	.064	**.346**	.078	.045	.000	.280	.047	.224	w2vec*	.130	.173	.251	.284	.000	.323	.127	.235
Smoothing with context	BM25-c	.567	.548	.043	.305	.405	.462	.338	.438	GloVe-p	.064	.274	.411	.373	.021	.349	.165	.332
	BM25-o	.641	.558	.000	.311	.399	.454	.347	.441	GloVe-r	.696	.585	.222	.227	.295	.501	.404	.438
	BM25-p	.785	.595	**.840**	.489	.255	.553	.627	.545	w2vec-c	.385	.422	.416	.503	.340	.432	.380	.452
	BM25-r	.557	.532	.472	.296	.530	.580	.520	.469	w2vec-o	.385	.427	.367	.483	.453	.441	.402	.450
	TF-ISF-c	.575	.528	.091	.303	.428	.458	.334	.430	w2vec-p	.064	.202	.314	.393	.000	.350	.126	.315
	TF-ISF-o	.575	.533	.000	.300	.428	.450	.334	.428	w2vec-r	.658	.570	.283	.321	.486	.542	.476	.478
	TF-ISF-p	.674	**.605**	.756	**.559**	.632	**.662**	**.687**	**.609**	fastt-c	.520	.478	.000	.360	.686	.488	.402	.442
	TF-ISF-r	.583	.491	.455	.368	**.761**	.654	.600	.504	fastt-o	.535	.471	.000	.288	.386	.451	.307	.403
	QLLM-c	.410	.497	.626	.516	.619	.481	.552	.498	fastt-p	.000	.150	.300	.379	.000	.329	.100	.286
	QLLM-o	.596	.492	.481	.499	.576	.459	.551	.483	fastt-r	**.818**	.592	.081	.196	.572	.545	.490	.444
	QLLM-p	.021	.360	.069	.042	.000	.284	.030	.229	w2vec*-c	.331	.379	.137	.356	.422	.409	.297	.381
	QLLM-r	.069	.390	.142	.085	.064	.413	.092	.296	w2vec*-o	.316	.387	.137	.374	.258	.377	.237	.379
	GloVe-c	.396	.448	.022	.272	.232	.374	.217	.365	w2vec*-p	.130	.173	.251	.284	.000	.323	.127	.235
	GloVe-o	.464	.457	.045	.248	.135	.377	.215	.361	w2vec*-r	.733	.553	.135	.199	.456	.532	.441	.428
Query expansion	BM25-g	.355	.286	.280	.463	.240	.360	.292	.370	GloVe-e	.171	.354	.000	.094	.260	.384	.144	.277
	BM25-e	.355	.413	.000	.025	**.590**	.439	.315	.292	w2vec-g	.202	.297	.335	**.571**	.397	.384	.311	.417
	TF-ISF-g	**.575**	**.477**	.076	.433	.398	.386	**.350**	**.432**	w2vec-e	.138	.330	.000	.221	.372	.404	.170	.318
	TF-ISF-e	**.575**	.467	.000	.021	.308	.385	.294	.291	fastt-g	.208	.368	.335	.517	.397	.359	.313	.415
	QLLM-g	.349	.310	**.394**	.494	.213	.343	.319	.382	fastt-e	.127	.347	.000	.073	.484	.383	.204	.268
	QLLM-e	.354	.316	.000	.073	.574	**.441**	.309	.277	w2vec*-g	.174	.278	.383	.553	.175	.343	.244	.391
	GloVe-g	.208	.349	.340	.484	.221	.368	.256	.400	w2vec*-e	.221	.307	.000	.004	.237	.391	.153	.234
Novelty	NewWords	**.504**	**.575**	.000	**.041**	.796	.556	**.433**	**.391**	w2vec-w	.432	.449	.000	.000	.600	.573	.344	.341
	NewWrdRatio	.376	.502	.000	.015	**.808**	**.624**	.395	.380	fastt-w	.109	.397	.000	.003	.529	.511	.213	.304
	GloVe-w	.043	.366	.000	.003	.566	.535	.203	.301	w2vec*-w	.358	.408	.000	.003	.557	.458	.305	.290
Compound	Tp-Tg	.807	.655	.756	.601	.611	.632	**.725**	.629	Qc-Tg	.622	.543	.626	.661	.619	.448	.622	.551
	Tp-Nr	.674	.609	.756	.586	.632	**.673**	.687	.623	Qc-Nr	.477	.581	.626	.520	**.806**	.543	.636	.548
	Tp-Tg-Nr	.807	.673	.756	.659	.611	.654	**.725**	**.662**	Qc-Tg-Nr	.706	.615	.626	**.705**	**.806**	.518	.713	.613
	Bp-Tg	.808	.636	**.840**	.594	.190	.534	.613	.588	Wr-Tg	.619	.620	.383	.406	.486	.504	.496	.507
	Bp-Nr	.785	.619	**.840**	.504	.319	.593	.648	.572	Wr-Nr	.827	.652	.349	.332	.565	.569	.580	.518
	Bp-Tg-Nr	.808	.679	**.840**	.629	.336	.574	.661	.627	Wr-Tg-Nr	**.830**	**.705**	.389	.438	.501	.548	.573	.564

This further corroborates the usefulness of the sentences' context in the task of retrieving sentences for statutory interpretation.

The similarity measuring methods from the TF-IDF family (i.e., BM25 and TF-ISF) mostly benefited from local contexts, especially the fixed paragraph context. For the remaining methods the paragraph context was the least advantageous. While the methods based on word embeddings performed best on the "soft" recursive context, which is on average two or three times as large as the paragraph context, the QLLM method was at its strongest on the whole case context. Since similar methods tend to prefer similar types of contexts we believe these preferences have an underlying cause. We leave investigation of this trend for future work.

The "context-aware" models perform surprisingly well, especially the two from the TF-IDF family. Apart from the issues these models are obviously incapable of solving (e.g., a lonely high value sentence in a case that does not deal with the term of interest), we noticed two systematic problems on IEV that presumably generalize to the other two queries:

(1) A verbatim citation of the source provision (or more frequently its part) that is included in a local context that heavily discusses the term of interest.
(2) Cases that frequently mention a term which is the same as the term of interest, but which comes from a different domain and as such is not useful for the interpretation of the term of interest.

By their very nature the methods operating on various contexts cannot deal with these two problems. While the methods operating on smaller local contexts are more susceptible to problem (1), the methods using larger contexts are more harmed by issue (2).

Table 3: Top 5 results for each query (Tp-Tg-Nr model)

Independent economic value	
H	[…] testimony also supports the independent economic value element in that a manufacturer could […] be the first on the market […]
H	[…] the information about vendors and certification has independent economic value because it would be of use to a competitor […] as well as a manufacturer
C	[…] the designs had independent economic value […] because they would be of value to a competitor who could have used them to help secure the contract
P	Plaintiffs have produced enough evidence to allow a jury to conclude that their alleged trade secrets have independent economic value.
C	Defendants argue that the trade secrets have no independent economic value because Plaintiffs' technology has not been "tested or proven."
Identifying particular	
H	In circumstances where duty titles pertain to one and only one individual […], duty titles may indeed be "identifying particulars" […]
P	Appellant first relies on the plain language of the Privacy Act which states that a "record" is "any item … that contains […] identifying particular […]
H	Here, the district court found that the duty titles were not numbers, symbols, or other identifying particulars.
P	[…] the Privacy Act […] does not protect documents that do not include identifying particulars.
H	[…] the duty titles in this case are not "identifying particulars" because they do not pertain to one and only one individual.
Common business purpose	
H	[…] the fact of common ownership of the two businesses clearly is not sufficient to establish a common business purpose.
P	Because the activities of the two businesses are not related and there is no common business purpose, the question of common control is not determinative.
H	It is settled law that a profit motive alone will not justify the conclusion that even related activities are performed for a common business purpose.
H	It is not believed that the simple objective of making a profit for stockholders can constitute a common business purpose […]
H	[…] factors such as unified operation, related activity, interdependency, and a centralization of ownership or control can all indicate a common business purpose.

Table 4: Importance of sentences' context (from 0 to 1)

	BM25	TF-ISF	QLLM	GloVe	w2vec	fastt	w2vec*
case	1.0	1.0	.6	.9	1.0	1.0	1.0
opinion	1.0	1.0	.7	.9	1.0	1.0	1.0
paragraph	1.0	.9	.1	.5	.2	.2	0.0
recursive	.6	.6	.6	.6	.6	.6	.6

We discovered that expanding the query with the words from the source provision or the most similar words generated by the trained word2vec model does not increase the performance of the models. With hindsight, this makes perfect sense in case of the expansion with the words from the source provision. Such a model prefers sentences that cite pieces of the provision. These are often the exact sentences we wish to avoid since they do not provide any additional information. As for expansion with the most similar words, the poor performance is baffling since the similar words provided by the word2vec model are appealing. Consider the top and bottom 3 from the 500 word long list for IEV:

> overall, *profitability*, *competitive*, ... *revenue*, projection, status

The length of the list is not the reason since we experimented with lists of different sizes.

Matching the expanded queries to the whole cases proved to be a more viable strategy. At least most of these models outperform the simple models that attempt to retrieve the sentences directly (Subsection 4.1). Yet, their counterparts from Subsection 4.2 perform better which suggests that the expansion of the query harms the model. Nevertheless, these models have an interesting property of preferring cases that discuss the source provision or the term of interest as opposed to the cases that mention a term equal to the term of interest that comes from a different domain. Thus, these models were able to mitigate problem (2) mentioned above.

Our experiments indicate that models focused on "novel" sentences (i.e., those that provide additional information) perform better than models focused on sentences that are most "similar" to the term of interest. We hypothesize that the models based on novelty detection benefit from focusing on one aspect of the relevance definition and handle that aspect rather well. By contrast, the models aimed at retrieving sentences on the basis of their similarity to the query suffer from the lack of overlap in words or meaning discussed earlier. In the experiments on novelty the models measuring the extra amount of words performed better than models that measure the semantic distance between a sentence and the provision. This makes sense. The relevance definition focuses on the presence of additional information, which is related to but somewhat different from documents being distant in terms of their meaning. The novelty detection models have an interesting property of identifying sentences that are complete or partial verbatim citations of the source provision. Thus, these models turned out to be capable of mitigating problem (1) mentioned earlier.

The interesting properties of the methods based on query expansion and novelty detection motivated us to propose the framework that deploy these models on top of the base ranker. The goal was to apply these methods in such a way as to help the base ranker deal with the two systematic problems that we identified (i.e., filter out cases from different domains and identify complete or partial citations of the source provision). The challenge was not to interfere with the base ranker too much since its ranking is vastly superior to that of the supporting methods. The initial attempts to connect the models through linear interpolation yielded poor results. The approach that turned out to be successful was to apply the supporting methods as indicator variables. These only fire in case of very strong evidence that one of the problems is present and effectively discard the problematic sentence. Otherwise, they do not have any effect on the order of sentences produced by the base ranker.

7 FUTURE WORK

We plan to significantly increase the size of the data set. In [35] the corpus consisting of only 243 sentences was used. That size was sufficient for the initial feasibility study. Here, we work with a data set of 4,635 sentences. This data set supported our experiments in identifying some of the challenges specific to sentence retrieval, as well as problems specific to the task of retrieving case-law sentences for statutory interpretation. The data set was also reasonable for evaluation of different methods with respect to which of them work well or have interesting properties. However, a data set of this size does not allow us to draw finer grained conclusions as to which of the methods perform better if their performance appears to be

quite similar. In addition, a larger data set would support a more principled optimization of methods that rely on hyperparameters.

In [35] it is shown that features such as a presence of a reference to the source provision, syntactic importance of the term of interest, structural placement of the sentence (such as its membership in a part as described in [37, 38]), or attribution [36, 40] are useful in predicting the value of a sentence (accuracy> .69). Integrating these methods into the proposed framework could increase the ranker's performance. In any case, a deeper semantic analysis of the sentences (perhaps, focused on finding typical patterns as in [41]) appears to be the most reasonable path forward.

8 CONCLUSIONS

We performed a detailed study on a number of retrieval methods in the context of the specialized task of retrieving case-law sentences for statutory interpretation. We confirmed that retrieving the sentences directly by measuring similarity between the query and a sentence yields mediocre results. Taking into account sentences' context turned out to be the crucial step in improving the performance of the ranking. We observed that query expansion and novelty detection techniques are able to capture information that could be used as an additional layer in a ranker's decision. Based on the detailed error analysis we integrated the context-aware ranking methods with the components based on query expansion and novelty detection into a specialized framework for retrieval of case-law sentences for statutory interpretation. Evaluation of different implementations of the framework shows promising results.

ACKNOWLEDGMENTS

This work was supported in part by a National Institute of Justice Graduate Student Fellowship (Fellow: Jaromir Savelka) Award # 2016-R2-CX-0010, "Recommendation System for Statutory Interpretation in Cybercrime," and by a University of Pittsburgh Pitt Cyber Accelerator Grant entitled "Annotating Machine Learning Data for Interpreting Cyber-Crime Statutes."

REFERENCES

[1] James Allan, Courtney Wade, and Alvaro Bolivar. 2003. Retrieval and novelty detection at the sentence level. In *Proc. of the 26th international ACM SIGIR conference on Research and development in informaion retrieval*. ACM, 314–321.
[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146.
[3] Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL* (2009), 31–40.
[4] Hang Cui, Min-Yen Kan, Tat-Seng Chua, and Jing Xiao. 2004. A comparative study on sentence retrieval for definitional question answering. In *SIGIR Workshop on Information Retrieval for Question Answering (IR4QA)*. 383–390.
[5] Jordan Daci. 2010. Legal Principles, Legal Values and Legal Norms: are they the same or different? *Academicus International Scientific Journal* 02 (2010), 109–115.
[6] Emile de Maat, Kai Krabben, Radboud Winkels, et al. 2010. Machine Learning versus Knowledge Based Classification of Legal Texts.. In *JURIX*. 87–96.
[7] Emile de Maat and Radboud Winkels. 2009. A next step towards automated modelling of sources of law. In *Proc. of the 12th ICAIL*. ACM, 31–39.
[8] Alen Doko, Maja Stula, and Darko Stipanicev. 2013. A recursive tf-isf based sentence retrieval method with local context. *IJMLC* 3, 2 (2013), 195.
[9] Timothy Endicott. 2000. *Vagueness in Law*. Oxford University Press.
[10] Timothy Endicott. 2014. Law and Language The Stanford Encyclopedia of Philosophy. http://plato.stanford.edu/. Accessed: 2016-02-03.
[11] Ronald T Fernández, David E Losada, and Leif A Azzopardi. 2011. Extending the language modeling framework for sentence retrieval to include local context. *Information Retrieval* 14, 4 (2011), 355–389.
[12] Alejandro Figueroa and John Atkinson. 2012. Contextual language models for ranking answers to natural language definition questions. *Computational Intelligence* 28, 4 (2012), 528–548.
[13] John Rupert Firth. 1957. A synopsis of linguistic theory 1930–1955. *Studies in Linguistic Analysis* (1957).
[14] Ingo Glaser, Elena Scepankova, and Florian Matthes. 2018. Classifying Semantic Types of Legal Sentences: Portability of Machine Learning Models. In *JURIX*.
[15] Zellig S. Harris. 1954. Distributional Structure. *WORD* 10, 2-3 (1954), 146–162. https://doi.org/10.1080/00437956.1954.11659520
[16] Herbert L. Hart. 1994. *The Concept of Law* (2nd ed.). Clarendon Press.
[17] Stefan Höfler, Alexandra Bünzli, and Kyoko Sugisaki. 2011. *Detecting legal definitions for automated style checking in draft laws*. Technical Report. Department of Informatics, University of Zurich.
[18] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651* (2016).
[19] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* (2016).
[20] Matjaz Juršic, Igor Mozetic, Tomaz Erjavec, and Nada Lavrac. 2010. Lemmagen: Multilingual lemmatisation with induced ripple-down rules. *Journal of Universal Computer Science* 16, 9 (2010), 1190–1214.
[21] Klaus Krippendorff. 2011. Computing Krippendorff's alpha-reliability. (2011).
[22] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on ML*. 957–966.
[23] D. N. MacCormick and R. S. Summers. 1991. *Interpreting Statutes*. Darmouth.
[24] C.D. Manning, P. Raghavan, and H. Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.
[25] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv:1301.3781* (2013).
[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[27] Tomas Mikolov, Scott Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations. In *Proc. of the 2013 Conference of the North American Chapter of the ACL: HLT*. ACL.
[28] Saeedeh Momtazi, Matthew Lease, and Dietrich Klakow. 2010. Effective term weighting for sentence retrieval. In *International Conference on Theory and Practice of Digital Libraries*. Springer, 482–485.
[29] Vanessa G Murdock. 2006. *Aspects of sentence retrieval*. Technical Report. Massachusetts University Amherst Department of Computer Science.
[30] María-Dolores Olvera-Lobo and Juncal Gutiérrez-Artacho. 2010. Question-answering systems as efficient sources of terminological information: an evaluation. *Health Information & Libraries Journal* 27, 4 (2010), 268–276.
[31] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.
[32] Jay M Ponte and W Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 275–281.
[33] The President and Fellows of Harvard University. 2018. Caselaw Access Project. https://case.law/. Accessed: 2018-12-21.
[34] Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, 45–50.
[35] Jaromir Savelka and Kevin D Ashley. 2016. Extracting case law sentences for argumentation about the meaning of statutory terms. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*. 50–59.
[36] Jaromír Savelka and Kevin D Ashley. 2017. Detecting Agent Mentions in US Court Decisions.. In *JURIX*. 39–48.
[37] J Savelka and Kevin D Ashley. 2017. Using conditional random fields to detect different functional types of content in decisions of united states courts with example application to sentence boundary detection. In *Workshop on Automated Semantic Analysis of Information in Legal Texts*.
[38] Jaromir Savelka and Kevin D. Ashley. 2018. Segmenting U.S. Court Decisions into Functional and Issue Specific Parts. In *JURIX*.
[39] Jaromir Savelka, Vern R Walker, Matthias Grabmair, and Kevin D Ashley. 2017. Sentence boundary detection in adjudicatory decisions in the united states. *Traitement automatique des langues* 58, 2 (2017), 21–45.
[40] Vern R Walker, Parisa Bagheri, and Andrew J Lauria. 2015. Argumentation Mining from Judicial Decisions: The Attribution Problem and the Need for Legal Discourse Models. In *Workshop on Automated Detection, Extraction and Analysis of Semantic Information in Legal Texts (ASAIL-2015)*.
[41] Stephan Walter. 2009. Definition extraction from court decisions using computational linguistic technology. *Formal Linguistics and Law* 212 (2009), 183.
[42] Bernhard Waltl, Florian Matthes, Tobias Waltl, and Thomas Grass. 2016. LEXIA: A data science environment for Semantic analysis of german legal texts. *Jusletter IT* 4, 1 (2016), 4–1.