Legal Knowledge and Information Systems
S. Villata et al. (Eds.)

doi:10.3233/FAIA200860

Sentence Embeddings and High-Speed Similarity Search for Fast Computer Assisted Annotation of Legal Documents

Hannes WESTERMANN [a,1], Jaromír ŠAVELKA [b], Vern R. WALKER [c], Kevin D. ASHLEY [d] and Karim BENYEKHLEF [a]

[a] *Cyberjustice Laboratory, Faculté de droit, Université de Montréal*
[b] *School of Computer Science, Carnegie Mellon University*
[c] *LLT Lab, Maurice A. Deane School of Law, Hofstra University*
[d] *School of Computing and Information, University of Pittsburgh*

Abstract. Human-performed annotation of sentences in legal documents is an important prerequisite to many machine learning based systems supporting legal tasks. Typically, the annotation is done sequentially, sentence by sentence, which is often time consuming and, hence, expensive. In this paper, we introduce a proof-of-concept system for annotating sentences "laterally." The approach is based on the observation that sentences that are similar in meaning often have the same label in terms of a particular type system. We use this observation in allowing annotators to quickly view and annotate sentences that are semantically similar to a given sentence, across an entire corpus of documents. Here, we present the interface of the system and empirically evaluate the approach. The experiments show that lateral annotation has the potential to make the annotation process quicker and more consistent.

Keywords. Annotation, Language Models, Sentence Embeddings, Approximate Nearest Neighbour, Interactive Machine Learning

1. Introduction

A lot of AI & Law research is enabled by annotation of legal texts. The annotation can be performed on several levels of textual units, such as the entire document, the paragraph, or an individual sentence. In this work, we focus on annotations performed on the sentence level. AI & Law research has employed a variety of annotation schemes on the sentence level, such as the annotation of:

- the rhetorical roles sentences play in a legal case (such as factual circumstances, a legal rule or an application of a legal rule to factual circumstances);
- the presence or absence of a certain factual circumstance the sentence describes (such as whether a security measure was present in a trade-secret case);
- the type and attributes of contractual clauses (such as the kind of liability addressed in a certain clause); and

[1] Corresponding Author: Hannes Westermann, E-mail: hannes.westermann@umontreal.ca

- the relevance of a sentence retrieved from a legal case to interpret a statutory term.

An annotated corpus of documents has many useful applications. For instance, a classification algorithm may be trained to infer labels for new sentences in a larger corpus of documents. This may lead, for example, to insights about the distribution of clauses in a large data set of contracts, to improved predictions about the outcome of a legal case from factors, or to ranking documents by relevance to a particular search query.

Typically, annotation of documents is performed by one or several annotators using a tool that allows them to review one document at a time, and to sequentially assign a label for each sentence as it occurs in that document. This approach has significant drawbacks. First, it is inefficient because annotating large corpora in this way takes a long time and is expensive. The label has to be determined from scratch for each sentence, causing significant cognitive overhead. Second, the annotations might be inconsistent across similar sentences. Since annotators often work through thousands of sentences, they may not remember how a certain sentence type was annotated the last time they saw it. If multiple annotators are involved, this problem may be exacerbated, as similar sentences are reviewed by different annotators.

In this paper, we investigate an alternative approach which we call "lateral annotation." Similarly to the traditional approach, annotators use a tool to view documents and label sentences. However, given any sentence there is an option to see sentences across the entire corpus (or from the rest of the document) that are semantically similar to the focused sentence. This feature uses sentence encoders based on deep learning models and libraries to quickly deliver semantically similar sentences using approximate nearest neighbour searches. The annotator then has the option of reviewing these similar sentences and assigning labels to one or more of them. Although the computer system assists the user by showing similar sentences, the choice of how to label a sentence ultimately rests with the annotator. It is therefore a hybrid approach, using machine learning to support human annotators with their task.

Legal language is often formalized and uses recurring linguistic structures. This means that identical, or very similar, sentences often appear in many documents. For example, contract clauses specifying a certain type of liability might often use the same words, syntax and sentence structure. Lateral annotation makes use of this attribute of legal language by allowing the annotator to label all similar sentences at one time. This can increase the speed of annotation. Since all similar sentences can be labelled at once, the consistency of the annotations is also likely to increase. Consequently this approach can significantly ease the important task of labelling large data sets in the legal domain.

2. Related Work

Branting et al. [4,5] proposed a semi-supervised approach to annotation of case decisions. The approach is based on several observations about the consistency of language across separate cases and within different sections of the same case. The researchers annotated a small set of decisions and calculated the mean of the semantic vectors [2,17] of all the spans annotated by a given tag (the "tag centroid"). The annotations were then projected to semantically similar sentences in the entire corpus to enable explainable prediction. In our work, we describe a hybrid method where we show the semantically similar sentences to an annotator for rapid and reliable annotation.

A steady line of work in AI & Law focuses on making the annotation effort more effective. Westermann et al. [26] proposed and assessed a method for building strong, explainable classifiers in the form of Boolean search rules. Employing an intuitive interface, the user develops Boolean rules for matching instead of annotating the individual sentences. Here, we replace the Boolean matching rules with sentence semantic similarity. Instead of developing the rules, the user confirms that the semantically similar sentences should be labeled as instances of the same types. Šavelka and Ashley [21] evaluated the effectiveness of an approach where a user labels the documents by confirming (or correcting) the prediction of a ML algorithm (interactive approach). The application of active learning has been explored in the context of classification of statutory provisions [25] and eDiscovery [8,9]. Hogan et al. [13] proposed and evaluated a human-aided computer cognition framework for eDiscovery. Tools to retrieve and rank text fragment by similarity for coding have further been implemented in qualitative data analysis tools, such as QDA Miner[2] and Nvivio.[3]

3. Proposed Framework

We investigate a system that enables an annotator to perform lateral annotations on a corpus of documents. We use sentence embeddings to capture the meaning of sentences, and then use approximate nearest neighbour search to find sentences that are semantically similar to a source sentence. This enables us to provide the annotators with viable sentence candidates for annotation in sub-second time.

3.1. Sentence Embeddings

In order to search for similar sentences based on an original sentence, we need a way to store sentences in a vector format that makes comparison easy. There are several ways of representing sentences in this way.

A bag of words representation (e.g., TF-IDF) is a simple but effective way to encode the meaning of a sentence. It has, however, at least two notable disadvantages: an enormously large feature space and the inability to account for the relatedness of meaning in different words. This means that sentences with the same meaning may be deemed to be completely different if they use largely non-overlapping vocabulary (e.g., synonyms). This is problematic for applications where sentence similarity is a key component (as in this work). In our experiments, we include the representation as a strong baseline due to its simplicity and wide usage.

More recently, pre-trained word embeddings and language models have gained popularity in creating word embeddings. These representations are motivated by the so-called distributional hypothesis: words that are used and occur in the same contexts tend to have similar meanings. [12] The idea that “a word is characterized by the company it keeps” was popularized by Firth. [11] The gist of the method is that words with similar meanings are projected onto similar vectors, by analyzing massive corpora of text to learn the distributions. There are several ways of combining these word vectors to produce sentence vectors, of which we have chosen three:

[2] provalisresearch.com/products/qualitative-data-analysis-software

[3] www.qsrinternational.com/nvivo-qualitative-data-analysis-software/home

1. Cer et al. [6] use the transformer architecture [22] and Deep Averaging Network [15] trained on the SNLI dataset. We work with the implementation released by the authors as the Google Universal Sentence Encoder (**GUSE**).[4]
2. Reimers et al. [19] build on top of BERT [10] and RoBERTa [18], which have been shown to be remarkably effective on a number of NLP tasks. Specifically, they use siamese and triplet network structures to derive semantically meaningful sentence embeddings. The authors released the models as Sentence Transformers (**ST**). We use this implementation in our work.[5]
3. Conneau et al. [7] demonstrate the effectiveness of models trained on a natural language inference task (Stanford NLI dataset [3]). They propose a BiLSTM network with max pooling trained with fastText word embeddings [2,17] as the best universal sentence encoding method. We adapt the implementation released by the authors which is commonly referred to as **InferSent**.[6]

3.2. *Efficient Similarity Search over High-dimensional Vectors*

Document search traditionally relies on a combination of relational databases built on structured data (metadata search) and inverted indexes (full-text search). These cannot deal efficiently with the vectors that represent documents' meaning. The brute-force approaches to indexing (i.e., store the results for all documents) or querying (i.e., compare the query to each data point) do not scale well beyond fairly small data sets.

In order to achieve semantic similarity search with the desired properties, one has to tackle the problem of preprocessing a set of n data points $P = \{p_1, p_2 \ldots, p_n\}$ in some metric space $\mathbb{X}$ (e.g., the d-dimensional Euclidean space $\mathbb{R}^d$) so as to efficiently answer queries by finding the point in P closest to a query point $q \in \mathbb{X}$. Solutions to this well-studied problem in other domains can be readily applied in our context. [14] For example, images and videos have become a massive source of data for indexing and search. And since it is often not practical to manually annotate the data to enable the use of relational databases, a search in some sort of a vector space remains the only option.

Johnson et al. [16] proposed a system that allows efficient indexing and search over collections containing around 1 billion documents. This shows that the current state-of-the-art is capable of supporting virtually any practical scenario in a legal annotation domain. For example, Šavelka et al. [20] segmented the whole corpus of US case-law into 0.5 billion sentences. While the technique in [16] would allow for efficient semantic similarity search over such a collection, it is most likely several magnitudes larger than any realistic legal annotation task. We utilize the Annoy similarity search library released by Spotify[7] for its ease of use and minimal system requirements. Annoy is an efficient implementation of the Approximate Nearest Neighbors algorithm proposed in [14].

3.3. *Lateral Annotation*

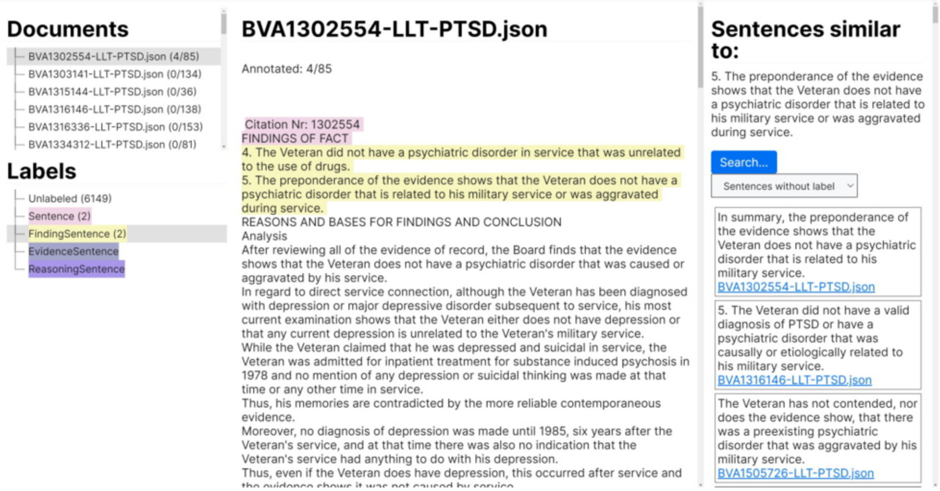
Semantic sentence embeddings and efficient vector similarity search are combined to enable lateral annotation. We have developed a prototype interface called CAESAR,

[4] https://tfhub.dev/google/universal-sentence-encoder/4

[5] github.com/UKPLab/sentence-transformers

[6] github.com/facebookresearch/InferSent

[7] github.com/spotify/annoy


Documents
BVA1302554-LLT-PTSD.json (4/85)
BVA1303141-LLT-PTSD.json (0/134)
BVA1315144-LLT-PTSD.json (0/36)
BVA1316146-LLT-PTSD.json (0/138)
BVA1316336-LLT-PTSD.json (0/153)
BVA1334312-LLT-PTSD.json (0/81)
Labels
Unlabeled (6149)
Sentence (2)
FindingSentence (2)
EvidenceSentence
ReasoningSentence
Search...
Sentences without label

BVA1302554-LLT-PTSD.json

Annotated: 4/85

Citation Nr: 1302554
FINDINGS OF FACT
4. The Veteran did not have a psychiatric disorder in service that was unrelated to the use of drugs.
5. The preponderance of the evidence shows that the Veteran does not have a psychiatric disorder that is related to his military service or was aggravated during service.
REASONS AND BASES FOR FINDINGS AND CONCLUSION
Analysis
After reviewing all of the evidence of record, the Board finds that the evidence shows that the Veteran does not have a psychiatric disorder that was caused or aggravated by his service.
In regard to direct service connection, although the Veteran has been diagnosed with depression or major depressive disorder subsequent to service, his most current examination shows that the Veteran either does not have depression or that any current depression is unrelated to the Veteran's military service.
While the Veteran claimed that he was depressed and suicidal in service, the Veteran was admitted for inpatient treatment for substance induced psychosis in 1978 and no mention of any depression or suicidal thinking was made at that time or any other time in service.
Thus, his memories are contradicted by the more reliable contemporaneous evidence.
Moreover, no diagnosis of depression was made until 1985, six years after the Veteran's service, and at that time there was also no indication that the Veteran's service had anything to do with his depression.
Thus, even if the Veteran does have depression, this occurred after service and the evidence shows it was not caused by service

Sentences similar to:

5. The preponderance of the evidence shows that the Veteran does not have a psychiatric disorder that is related to his military service or was aggravated during service.

In summary, the preponderance of the evidence shows that the Veteran does not have a psychiatric disorder that is related to his military service.
BVA1302554-LLT-PTSD.json

5. The Veteran did not have a valid diagnosis of PTSD or have a psychiatric disorder that was causally or etiologically related to his military service.
BVA1316146-LLT-PTSD.json

The Veteran has not contended, nor does the evidence show, that there was a preexisting psychiatric disorder that was aggravated by his military service.
BVA1505726-LLT-PTSD.json

Figure 1. A screenshot of the prototype Computer-Assisted Efficient Semantic Annotation & Ranking (CAESAR) Interface

Computer-Assisted Enhanced Semantic Annotation & Ranking, to demonstrate this capability. The sentence embedding frameworks are used to create semantic embeddings of all sentences in a corpus. These are then used to create an index for fast similarity search.

The capability is provided to the user through an annotation interface (see Figure 1). The interface allows the annotator to define a schema of labels in a hierarchical structure (i.e. a type system), and to tag individual sentences with these labels. For each sentence, it is possible to retrieve semantically similar sentences, using the methods described above. These are shown in the sidebar to the right, sorted by similarity in descending order. This panel allows the annotator to perform lateral annotations by quickly annotating the sentences shown, or to see the context of the shown sentences before annotation.

We envision the following procedure for labeling sentences using CAESAR. An annotator starts with the first document, and labels the first sentence. Then, he asks to be shown similar sentences in the sidebar. He then labels sentences in the sidebar until sentences are no longer similar enough to quickly allow the annotator to determine that they are of the same class. The annotator then returns to the full text of the case and labels the next sentence. As the annotator moves to the next documents, many of the sentences may already be labelled, and can be skipped.

This method of improving annotation efficiency is completely unsupervised. It can be implemented before having started any kind of annotation, by relying on the sophisticated neural models trained on huge datasets of general texts (e.g., news corpora, Wikipedia). Despite this, the method seems to perform very well on legal annotation tasks, as demonstrated in Section 4.

4. Evaluation

4.1. Datasets

In order to evaluate the lateral annotation method, we use three existing data sets:

1. Walker et al. [23] analyzed 50 fact-finding decisions issued by the U.S. Board of Veterans' Appeals (**BVA**) from 2013 through 2017, all arbitrarily selected cases dealing with claims by veterans for service-related post-traumatic stress disorder (PTSD). For each of the 50 BVA decisions in the PTSD dataset, the researchers extracted all sentences addressing the factual issues related to the claim for PTSD, or for a closely-related psychiatric disorder. These were tagged with the rhetorical roles the sentences play in the decision [24]. We conducted our experiments on this set of 6,153 sentences.[8]
2. Šavelka et al. [20] studied methods for retrieving useful sentences from court opinions that elaborate on the meaning of a vague statutory term. To support their experiments they queried the database of sentences from case law that mentioned three terms from different provisions of the U.S. Code. They manually classified the sentences in terms of four categories with respect to their usefulness for the interpretation of the corresponding statutory term. In [20] the goal was to rank the sentences with respect to their usefulness; here, we classify them into the four value categories (**StatInt**).[9]
3. Bhattacharya et al. [1] analyzed 50 opinions of the Supreme Court of India. The cases were sampled from 5 different domains in proportion to their frequencies (criminal, land and property, constitutional, labor and industrial, and intellectual property). From each of the 50 decisions the sentence boundaries were detected using an off-the-shelf general tool.[10] Then the 9,380 sentences were manually classified into one of the seven categories according to the rhetorical roles they play in a decision. Our experiments were conducted on this set of sentences (**IndSC**).[11]

4.2. Experiments

In order to evaluate the effectiveness of lateral annotation and compare different embedding methods, we use our system to retrieve the closest sentences to a query sentence, and investigate how many of them have the same label as the source sentence. Assuming that lateral annotation is more efficient than sequential annotation, the more retrieved sentences that have the same label, the more efficiently the annotator will be able to annotate the data set.

We report several metrics. First, we investigate the length of chains of annotations by traversing the retrieved sentences, from the most similar to the least, until we arrive at a label that does not match the label of the source sentence. We calculate the longest encountered chain (Max) and the average chain length (Avg) for each data set and em-

[8] Dataset available at `github.com/LLTLab/VetClaims-JSON`

[9] Dataset available at `github.com/jsavelka/statutory_interpretation`

[10] `spacy.io`

[11] Dataset available at `github.com/Law-AI/semantic-segmentation`

bedding method. Second, we determine how many of the top 20 closest sentences have the same label as the source sentence (P@20) - a measure of precision.

Third, we visualize the high-dimensional GUSE embeddings of all sentences in a dataset, reduced to 2 dimensions using a Principal Component Analysis. The colors in the resulting visualization correspond to the gold standard labels for the individual sentences. The most important feature of an embedding space for our purpose is that each sentence should be surrounded by multiple sentences with the same label, e.g. the same color.

4.3. Results

Table 1 presents the Max, Avg, and P@20 statistics for each data set and for each embedding method. Overall, the sentence embeddings seem to capture enough linguistic information to achieve improvement in all three metrics, without any training on the domain-specific data sets. The neural models perform much better than the random baseline, and perform better or equal to the TF-IDF baseline.

Looking at the individual data sets, it seems like the **Board of Veterans' Appeals data set** benefits significantly from lateral annotation. On average, 70% of the closest 20 sentences to each sentence have the same label, meaning an annotator can likely annotate these sentences laterally. This could offer a significant speed-up in the annotation of such a data set.

Looking at the individual labels in Table 2, the "Citation" label seems by far the easiest to annotate laterally. 94% of the top 20 closest sentences to a citation sentence are also citation sentences. This is likely due to the special tokens (such as parentheses, year numbers and special words such as "See") in these sentences. "Rule" also performs very well, which might be due to the same rule being cited in multiple cases. The embeddings capture these distinctions well, which can be seen in Figure 2, where sentences of the same type seem clearly concentrated in certain areas.

In the **Statutory Interpretation data set** the technique appears most suitable for the sentences labeled as "No value." This makes sense since these are mostly sentences that fully or partially quote or paraphrase a statutory provision. Hence, these sentences are often very similar to each other. The middle graph in Figure 2 confirms this observation. Three compact red clusters clearly correspond to the "No value" sentences associated with the three terms of interest. The sentences with the other three labels are somewhat more challenging. Yet, even for the more challenging categories, a significant amount of sentences could still be annotated laterally, as seen in Table 2.

The **Indian Supreme Court data set** is where lateral annotation gives the least advantage, with our models retrieving under 40% of matching sentences in the top 20 positions. On average, each sentence seems to be next to only 2.1-2.4 sentences of the same class in the embedding space. This can also be seen in the PCA visualization in figure 2. Unlike the other data sets, the sentence embeddings do not seem to result in clearly separate classes. The comparative difficulty of separating this data set might be explained by the fact that the sentences are selected from five different domains, and assigned seven labels—more than the other two data sets. The "Ratio" and "Facts" sections seem slightly easier to annotate in a lateral fashion, which might be due to a consistent structure or content of these sentences. It is surprising that the "Ratio" class has a high precision, while the "Ruling of lower court" has low precision, but this matches the findings of the authors in [1] for classification difficulty.

	Statistics								
	BVA			StatInt			IndSC		
	Max	Avg	P@20	Max	Avg	P@20	Max	Avg	P@20
Random	9	1.35	.24	18	2	.45	10	1.36	.24
TF-IDF	195	13.73	.59	197	24.40	.70	27	2.16	.36
GUSE	696	55.92	**.70**	257	25.80	**.73**	**45**	**2.43**	.37
ST	710	48.62	.68	**277**	**30.16**	.69	30	2.22	.35
InferSent	**863**	**83.92**	**.70**	204	22.5	.66	**45**	2.41	**.38**

Table 1. Statistics for different sentence embedding methods, including evaluation of chains of lateral annotation (Max, Avg) and how many of the 20 closest sentences on average have the same label (P@20).

BVA		SID		ISC	
Label	P@20	Label	P@20	Label	P@20
Sentence	0.60	No Value	0.89	Facts	0.40
Finding	0.49	Potential	0.58	Ruling (lower)	0.08
Evidence	0.76	Certain	0.15	Argument	0.18
Rule	0.73	High	0.33	Ratio	0.46
Citation	0.94			Statute	0.25
Reasoning	0.27			Precedent	0.32
				Ruling (present)	0.32

Table 2. The ratio of matching labels among the top 20 most similar sentences, per label

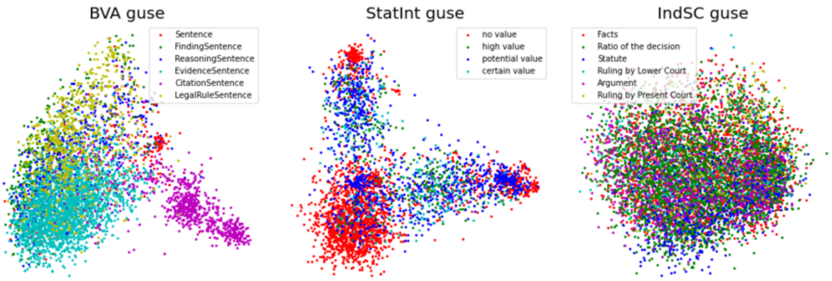


Figure 2. Visualizations of the sentences across the data sets, embedded using the GUSE and reduced to two dimensions using a Principal Component Analysis. The colors correspond to different labels.

5. Discussion

We have introduced and evaluated a lateral annotation framework. We experimented with four types of sentence embeddings and compared them against the random baseline. All of the embeddings showed significant improvements in selecting sentences that are of the same class compared to the random baseline. The neural models show strong performance across the three data sets. In general, they perform similarly, although the Google Universal Sentence Encoder and InferSent seems to have a slight edge. In the BVA data set, the neural models clearly outperform the TF-IDF baseline, while the performance is more balanced in the StatInt and IndSC data sets. Even in the most challenging data set, almost 40% of the 20 closest sentences had the same label as a source sentences. For the other data sets, this number was 70%. This indicates a significant potential benefit for using lateral annotation.

Different areas might benefit from the use of lateral annotations. The assumption behind the method is that sentences that have similar semantic embeddings are likely to belong to the same class that an annotator is aiming to label. This should work better for labeling schemas and data sets where the semantic properties of a sentence are linked to its label, and where the homogeneity of sentences with the same label is high. This can be seen in the per-class analysis of precision, showing that citation sentences and recitation of previous rules and cases are more suitable for lateral annotation. Sentences with less inherent structure and similarity, such as reasoning sentences, seem to perform slightly worse. The Indian Supreme Court data set, which draws from five different domains and uses seven classes, performs worse in a lateral annotation context, which could indicate that more diverse data sets are more difficult to annotate laterally.

Further, the method benefits from data sets where the set of sentences with a particular label is made up of several clusters of semantically similar sentences that the annotator can efficiently scope. For each sentence that is part of such a cluster, the annotator can efficiently label a large number of sentences. Outlier sentences, which do not belong to any larger cluster of sentences with the same label, are less likely to benefit from the method, as they do not assist the annotator in finding other sentences of the same label.

We hypothesize that the legal domain is well-suited for the lateral annotation method. Legal practitioners often use stereotypical language to describe certain facts, including a shared vocabulary and sentence structure. This shared language is more likely to be suited for annotation supported by semantic similarity search, and could significantly speed-up annotating large data sets with per-sentence labels.

6. Future Work

There are multiple ways of building upon this research. First, it is important to investigate how lateral annotation performs in additional real-world scenarios, and compare it to traditional methods of annotation. Second, finding ways to expand our framework by extending vectors with relevant properties or by combining vectors could increase the system's performance. Third, augmenting the method to integrate active learning approaches (where a machine learning model suggests which sample to label next to the annotator) could help to discover more relevant sentences. Furthermore, the approach of annotating sentences laterally could be used at an earlier stage, to support the exploration of data sets and the creation of type systems.

7. Conclusions

In this paper, we have explored a method for the efficient annotation of sentences, by leveraging sophisticated sentence embedding models and approximate nearest neighbour searches. Using these technologies, we designed a method and an interface that allow annotators to label similar sentences in one go across documents, rather than having to episodically label similar sentences as they come up in new documents. We investigated some properties of different possible embeddings and demonstrated the benefits of using the method on three legal data sets.

References

[1] Bhattacharya, P., S. Paul, K. Ghosh, S. Ghosh, and A. Wyner. “Identification of Rhetorical Roles of Sentences in Indian Legal Judgments.” arXiv preprint arXiv:1911.05405 (2019).

[2] Bojanowski, P., E. Grave, A. Joulin, and T. Mikolov. “Enriching word vectors with subword information.” *Transactions of the Association for Computational Linguistics* 5 (2017): 135-146.

[3] Bowman, S., G. Angeli, C. Potts, and C. Manning. “A large annotated corpus for learning natural language inference.” *arXiv preprint arXiv*:1508.05326 (2015).

[4] Branting, L. K., C. Pfeifer, B. Brown, L. Ferro, J. Aberdeen, B. Weiss, M. Pfaff, and B. Liao. “Scalable and explainable legal prediction.” Artificial Intelligence and Law (2020): 1-26.

[5] Branting, L.K., B. Weiss, B. Brown, C. Pfeifer, A. Chakraborty, L. Ferro, M. Pfaff, and A. Yeh. “Semi-supervised methods for explainable legal prediction.” In *Proc. ICAIL 2019*, pp. 22-31. 2019.

[6] Cer, D., Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. St John, N. Constant et al. “Universal sentence encoder.” *arXiv preprint arXiv*:1803.11175 (2018).

[7] Conneau, A., D. Kiela, H. Schwenk, L. Barrault, and A. Bordes. “Supervised learning of universal sentence representations from natural language inference data.” *arXiv preprint arXiv*:1705.02364 (2017).

[8] Cormack, G., and M. Grossman. “Scalability of continuous active learning for reliable high-recall text classification.” In Proc. 25th ACM Int’l Conf. on Info. & Knowledge Management, pp. 1039-1048. 2016.

[9] Cormack, G., and M. Grossman. “Autonomy and reliability of continuous active learning for technology-assisted review.” *arXiv preprint arXiv*:1504.06868 (2015).

[10] Devlin, J., M. Chang, K. Lee, and K. Toutanova. “Bert: Pre-training of deep bidirectional transformers for language understanding.” *arXiv preprint arXiv*:1810.04805 (2018).

[11] Firth, J. “A synopsis of linguistic theory, 1930-1955.” *Studies in linguistic analysis* (1957).

[12] Harris, Z. “Distributional structure.” *Word* 10, no. 2-3 (1954): 146-162.

[13] Hogan, C., R. Bauer, and D. Brassil. “Human-aided computer cognition for e-discovery.” In *Proc. 12th Int’l Conf. on Artificial Intelligence and Law*, pp. 194-201. 2009.

[14] Indyk, P., and R. Motwani. “Approximate nearest neighbors: towards removing the curse of dimensionality.” In *Proc. 30th Annual ACM Symposium on Theory of Computing*, pp. 604-613. 1998.

[15] Iyyer, M., V. Manjunatha, J. Boyd-Graber, and H. Daumé III. “Deep unordered composition rivals syntactic methods for text classification.” In *Proc. 53rd ann. meet. ACL* (Vol 1), pp. 1681-1691. 2015.

[16] Johnson, J., M. Douze, and H.é Jégou. “Billion-scale similarity search with GPUs.” IEEE Transactions on Big Data (2019).

[17] Joulin, A., E. Grave, P. Bojanowski, and T. Mikolov. “Bag of tricks for efficient text classification.” *arXiv preprint arXiv*:1607.01759 (2016).

[18] Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. “Roberta: A robustly optimized bert pretraining approach.” *arXiv preprint arXiv*:1907.11692 (2019).

[19] Reimers, N., and I. Gurevych. “Sentence-bert: Sentence embeddings using siamese bert-networks.” *arXiv preprint arXiv*:1908.10084 (2019).

[20] Šavelka, J., H. Xu, and K. Ashley. “Improving Sentence Retrieval from Case Law for Statutory Interpretation.” *Proc. 17th Int’l Conf. on Artificial Intelligence and Law*, pp. 113-122. 2019.

[21] Šavelka, J., G. Trivedi, and K. Ashley. “Applying an interactive machine learning approach to statutory analysis.” In Proc. 28th Ann. Conf. on Legal Knowledge & Info. Systems (JURIX’15). IOS Press. 2015.

[22] Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, Ł. Kaiser, and I. Polosukhin. “Attention is all you need.” In *Advances in neural information processing systems*, pp. 5998-6008. 2017.

[23] Walker, V., K. Pillaipakkamnatt, A. Davidson, M. Linares, and D. Pesce. “Automatic Classification of Rhetorical Roles for Sentences: Comparing Rule-Based Scripts with Machine Learning.” In ASAIL@ ICAIL. 2019.

[24] Walker, V., J. Han, X. Ni, and K. Yoseda. “Semantic types for computational legal reasoning: propositional connectives and sentence roles in the veterans’ claims dataset.” *Proc. ICAIL ’17*. ACM, 2017.

[25] Waltl, B., J. Muhr, I. Glaser, G. Bonczek, E. Scepankova, and F. Matthes. “Classifying Legal Norms with Active Machine Learning.” In JURIX, pp. 11-20. 2017.

[26] Westermann, H., J. Šavelka, V. Walker, K. Ashley, and K. Benyekhlef. “Computer-Assisted Creation of Boolean Search Rules for Text Classification in the Legal Domain.” In *JURIX*, pp. 123-132. 2019.