COLIEE 2020: Legal Information Retrieval & Entailment with Legal Embeddings and Boosting[⋆]

Houda Alberts[0000−0001−6924−5995], Akin Ipek[0000−0002−1363−5254], Roderick Lucas[0000−0002−5689−2197], and Phillip Wozny[0000−0001−9417−7170]

Deloitte NL, tax-i team, Amsterdam, Netherlands
{hdevries-alberts,aipek,rlucas,pwozny}@deloitte.nl

Abstract. In this paper we investigate three different methods for several legal document retrieval and entailment tasks; namely, new low complexity pre-trained embeddings, specifically trained on documents in the legal domain, transformer models and boosting algorithms. Task 1, a case law retrieval task, utilized a pairwise CatBoost resulting in an F1 score of .04. Task 2, a case law entailment task, utilized a combination of BM25+, embeddings and natural language inference (NLI) features winning third place with an F1 of 0.6180. Task 3, a statutory information retrieval task, utilized the aforementioned pre-trained embeddings in combination with TF-IDF features resulting in an F2 score of 0.4546. Lastly, task 4, a statutory entailment task, utilized BERT embeddings with XGBoost and achieved an accuracy of 0.5357. Notably, our Task 2 submission was the third best in the competition. Our findings illustrate that using legal embeddings and auxiliary linguistic features, such as NLI, show the most promise for future improvements.

Keywords: Legal Information Retrieval · Textual Entailment · Classification · Natural Language Inference · Ranking · Legal Embeddings · BERT · Boosting

1 Introduction

Search engines have become the gateway to the internet for both the layman and the scholar alike [27]. Google, the largest search engine by market share [6], has become an indispensable tool for academic researchers [36]. However, legal researchers have unique needs that require unique search tools [1]. As the methods that search engines use to rank results shape how the user interacts with web content [12], it is critical that legal researchers have access to tools designed with them in mind. Given the size of corpora that legal researchers must search through, any methods to increase the efficiency of legal research

[⋆] Supported by the tax-i team within Deloitte. Each author contributed equally and is responsible for one specific task; In order of the authors, responsible for task 3, 1, 2 and 4.

have disruptive potential for the industry [7]. The use of machine learning in the legal domain has the potential to reduce the amount of time required for legal research and reasoning [7]. Within tax-i [1], we develop machine learning tools built with legal researchers in mind, offering specialized search functionalities such as cross-lingual search.

Two main machine learning applications in the legal domain are entailment and information retrieval [[7], [35]]. Entailment aims to address the question of whether a given given proposition is true or false based on a piece of evidence [28]. Machine learning systems can then reason over entailed claims [24]. Information retrieval involves searching through a corpus of documents and ranking them according to their relevance to a query [19]. These are only two examples of how machine learning can disrupt the legal industry through the automation of costly, yet repetitive tasks [13].

As a means of cultivating research in these domains University of Alberta, along with a host of co-sponsors, hosts the Competition on Legal Information Extraction/Entailment (COLIEE). In its current iteration, COLIEE comprises four tasks. Task one requires identifying the set of cases from a case law corpus which support the decision of a query case. Given the decision fragment of a case that is supported by another case, task two aims to discern which paragraph of the supporting case entail the fragment. Tasks three and four use statutory data and bar exam questions. Task three involves retrieving statutory articles relevant to a bar exam question. Task four aims to determine the entailment of a bar exam question by a set of relevant articles.

1.1 Contributions

Our aims in COLIEE are the following:

- We intend to employ the state of the art natural language processing (NLP) methods to address the four tasks.
- Second, we address the fact that many state of the art language models do not transfer well to the legal domain. We do this by training our own embeddings on legal text.

The rest of the paper is as follows: Section 2 continues with the related work and Section 3 explains our novel approach with our legal embeddings. Next, in Section 4, 5, 6, and 7 we will focus on the methodology, experimental set-up and results for task 1, 2, 3 and 4 respectively. Finally, Section 8 concludes our paper with possible extensions for our research.

2 Related Work

In this section, we will focus on related research on both legal information retrieval and legal entailment. Although both serve different purposes, they can be addressed using common methods.

[1] https://tax-i.deloitte.nl/

Classic information retrieval techniques include BM-25 [23] and TF-IDF [18], which obtain normalized bag of word representations of a corpus of documents and a query. The aforementioned are then compared to rank the queried documents by relevancy. Such practices are common in the legal domain [[14], [10], [32]].

Richer document representation methods introduced in legal information retrieval include, Doc2Vec [16] and more advanced transformer methods [31], such as BERT [8]. One shortcoming of BERT is that it takes a fixed sequence length of 512 tokens, which is problematic given the length of legal documents [2]. Previous attempts to address this shortcoming have involved using summarization tools such as Gensim [25] or trimming sequences to the maximum length [10]. Another downside of out-of-the-box BERT is its inability to generalize to specific domains due to the fact that it was trained on Wikipedia-like data [11].

BERT can also be used only as means of generating features from text without invoking the entire transformer architecture [8]. Common methods of using BERT derived features include taking the mean of the last four hidden layers, taking a weighted sum of the last four hidden layers, or just using the last hidden layer [8]. BERT derived features can then be used as input to a separate model [10].

Both information retrieval and entailment tasks can be configured as classification problems through pairwise relevance prediction and binary classification, respectively. Legal classification problems can be addressed through deep learning methods. For instance, Chalkidis, Androutsopoulos & Aletras (2019) employed a Bi-Directional Gated Recurrent Unit with Attention (Bi-GRU-ATT) model [4] to predict the outcome of European Court of Human Rights (ECHR) cases. Bi-GRU-ATT models are useful in the legal domain because they are sensitive to context [3]. Previous COLIEE submissions have illustrated the effectiveness of encoding entailment inputs into separate Long Term Short Term Memory (LSTM) models whose combined output is used for binary classification [32].

Non deep learning methods can also be used for classification, such as K-Nearest Neighbor, Random Forest, and boosting algorithms [25]. The downside of this latter problem framing is that class imbalanced become more common due to more non-relevant documents; however, tree-based approaches such as XGBoost [5] or CatBoost [9] tend to handle such imbalances better. XGBoost is therefore often the model of choice in competitive machine learning environment [20]. Furthermore, tree based methods can make use of meta information, such as the date or header, alongside textual features [34].

3 Legal Embeddings

As explained in Section 2, most competitive embedding based models are trained on a general corpus, such as Wikipedia or (short) stories. However, when applied

[2] https://eur-lex.europa.eu/legal-content/NL/ALL/?uri=CELEX%3A61996CJ0349

to legal data, the results fall short. Legal English is different than regular English with respect to syntax, semantics, vocabulary and morphology, which explains this shortage in performance [33]. Based on these findings, we trained a legal FastText model [3] at the start of this year for our own applications within our intelligent information retrieval system, tax-i. The need for legal embeddings is verified by recent research that has found that training a legal BERT does aid in legal-based entailment and question answering [11].

To train our FastText [2] model, we make use of a partition of legal data that we have available in our tax-i platform. We only use US-related content and take roughly 1/3rd of this which translates to 1M US cases. We apply the pre-processing on this as needed by our models (i.e. lowercase, punctuation removal etc.) and use this during training.

We then use the unsupervised FastText [4] to train embeddings with the legal data via a skipgram model, training for 4 epochs, 6 threads, no wordngrams, a 300-dimensionality for the word embeddings and all remaining parameters remain default. This yields a final loss of around ~ 0.05. The legal embeddings resulted in sub-par performance on tasks one and two during initial experimentation; therefore, alternative methods were employed.

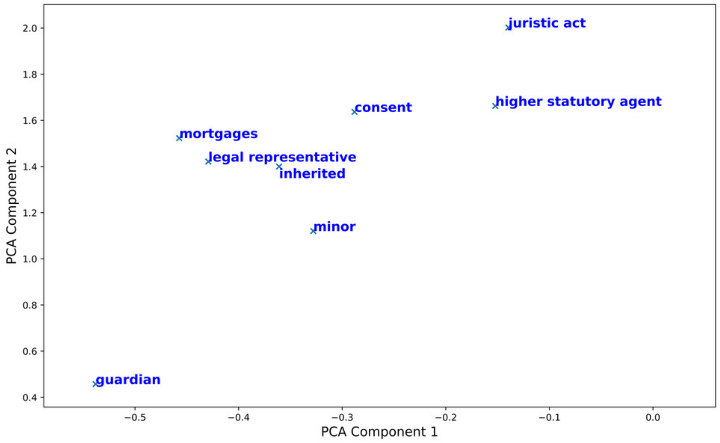

Fig. 1. Our legal embeddings visualized for several legal terms in a 2D space with the use of PCA.

When we visualize different legal terms with our legal embeddings, we can distinguish related and non-related legal concepts. Our intuition of these words corresponds with the visualization; we know that legal representatives are important when dealing with inheritance and mortgages. We can also observe that a legal representative and higher statutory agent are more similar than a guardian,

[3] We opted for FastText rather than BERT due to limited computational resources.

[4] https://github.com/facebookresearch/fastText

which have different roles to a person, showing that these legal embeddings show an understanding of the legal corpus.

While these short concepts contain rich information in their embeddings, we have observed that they do not attribute to improved performance in task 1 and 2 compared to task 3 and 4. This is likely due to the fact that, while these embeddings can be (mathematically) combined, the longer the sequence the less informative these embeddings become. Since the first two tasks involve large pieces of text, they struggle to obtain a balanced combination of the separate word embeddings. Therefore, legal embeddings were not used for task 1 and 2.

4 Task 1: Case Law Information Retrieval

The first task focuses on case law information retrieval where given a case law document, Q, we want to retrieve relevant case law to this document from a finite set of candidates $S_1, S_2, ...S_n$.

4.1 Methodology

We formulate this retrieval task into a pairwise classification problem meaning, where each candidate is labeled as relevant or not. We use English case law and have a fixed amount of candidates per case, namely 200, where the candidates differ for each base case.

To classify each candidate, we conjoin the query document and a summary of the candidate case, extracted using either Gensim or regex, to generate TF-IDF with IDF weight smoothing and absolute word count features using uni- and bigrams. These features are then put into a boosting algorithm, either XGBoost [5] or CatBoost [22], where the latter has shown promising results in text-related tasks [[26], [29]]. Afterwards, we use precision, recall and the F1 measure to assess how this model performs by only taking the ones classified as relevant into account for these measures. That is, we do not take the non-relevant cases into account to get a proper estimation of the classification in this retrieval task.

4.2 Experimental Setup

We use a 90-10% split on the training data to obtain a training and validation set. Using absolute counts and TF-IDF features with XGBoost and CatBoost resulted in F1 scores of 0.37 and 0.54, respectively. As such, we determined the superior method to be CatBoost with 1500 iterations, a learning rate of 0.1, l2 leaf regularization of 3.5, and depth of 4.

4.3 Results

Among the COLIEE 2020 submissions this year the used method, absolute counts and TF-IDF into CatBoost, we obtain a test F1 score of 0.0457, where the top-3 submissions obtain 0.6774 (team cyber), 0.6768 (team cyber) and 0.6682 (team TLIR) in descending order. The difference between the final test results and the training data test results was due to human error in the method of document summarization employed. That is, the model was trained on Gensim summarized documents and the model was evaluated on extracted summary documents via regular expressions. Applying Gensim summarization to the 2020 test data resulted in an actual F1 score of .18. The drop in performance is indicative of overfitting. This can be caused by the use of a pairwise TF-IDF approach which depends on a shared vocabulary between train and test data. This can be overcome by using the cosine distance between TF-IDF vectors as a feature. Furthermore, repeated evaluation on the training data showed high variance model performance with values between 0.43 and 0.54.

5 Task 2: Case Law Entailment

The case law entailment task requires finding an entailing paragraph from a case, given a base case with a specified fragment. Or put formally: Given a base case, Q, its entailed fragment, f, and another related case, R, composed of the paragraph set $P = \{p_1, p_2, \ldots, p_n\}$, find the paragraph set $E = \{p_1, p_2, \ldots, p_m \mid p_i \in P\}$, such that p_i entails f.

5.1 Methodology

The three feature groups are ensembled in a XGBoost classifier in order to predict the probability of p_i entailing f. XGBoost was chosen for the same reason that it is often the model of choice in competitive machine learning environments: it is robust to the curse of dimensionality, is performs feature selection automatically, and is capable of generating rich feature representations [20].

The feature groups are classical features, embedding similarity and Natural Language Inference (NLI).

Feature Group 1 - Classical Features These consist of classical word features, such as length of the paragraph p_i, place of paragraph within the article, count of overlapping words between f and p_i and a BM-25 ranking.

Feature Group 2 - Embedding Similarity This consists of the similarity of the means of three different word-embeddings: RoBERTa [17], BERT [8] and a fine-tuned version of the latter all uncased on the COLIEE 2020 training data. As a similarity measure between the embedded p_i and the embedded f, we use the cosine similarity.

Feature Group 3 - Natural Language Inference NLI is the task of determining whether two statements entail or contradict each other. Ideally this would be done utilizing a fine-tuned model for the legal domain. However, due to the training and validation time constraints, we use the pre-trained BERT NLI model.

Ensemble model The previous mentioned feature groups are all min-max normalized over all paragraphs in each related case R. Predictions are the probabilistic output of a XGBoost classifier. For every base case Q, we check how many paragraphs p_i of related case R are predicted to be entailing fragment f. If no paragraphs are found for a base case, the algorithm picks the one with the highest probabilistic outcome (albeit it with a low probability). If more than two are predicted, only the two with the highest scores are allowed in the submitted results.

5.2 Experimental Setup

After experimentation it was found that fine-tuned BERT embeddings provided the best performing similarity measure for feature group 2. As such, all approaches used fine-tuned BERT embeddings for feature group 2 and fed features into an XGBoost classifier. The three different approaches based on the aforementioned feature groups were the following. The Feature Importance approach used only features deemed important by analysing xgboost feature weights. The XGBoost approach used all features with a grid search hyperparameter tuning. Finally, the XGBoost Bayesian used all features with bayesian optimization hyperparameter tuning. We cross validate the training set with five folds, and our results can be found in Table 1. It is evident that we try to optimize precision over recall for more precise results.

Approach	Precision	Recall	F1
Feature Importance	0.5990	0.5800	0.5855
XGBoost	0.6168	0.5855	0.5984
XGBoost Bayesian	0.7054	0.4785	0.5674

Table 1. Validation results of the models for task 2 using 5-fold cross validation.

5.3 Results

Applying the previous mentioned models on the official COLIEE 2020 test set, we get the results as described in Table 2. All of the models achieve F1 scores high enough for the top seven submissions this year. Using Bayesian tuning, we obtain the best results from our own submissions and obtain third place, showing the importance of proper tuning.

Analysis Upon further analysis of the features, we can see that using the most important features is necessary, but not sufficient for state-of-the-art results. Furthermore, grid search hyperparameter tuning improves performance, but not as much as Bayesian optimization methods.

Team	F1
JNLP BMWT	0.6753
JNLP BMW	0.6222
TAXI XGBoost Bayesian	0.6180
TLIR	0.6154
JNLP WT	0.6094
TAXI XGBoost	0.5992
TAXI Feature Importance	0.5917

Table 2. The top seven F1 scores of COLIEE 2020 task 2.

Upon inspecting the test result data and XGBoost Bayesian predictions, we found the following to be true. The model performs better on longer candidate paragraphs. The average paragraph lengths for correctly and incorrectly predicted examples were 108 and 91 tokens, respectively. This suggests that the longer the paragraph the more BERT is capable of comparing contextual information. With respect to shared vocabulary, the model seems to perform better when shared vocabulary is slightly higher. The count of number of shared tokens in correctly and incorrectly predicted examples were 122 and 111, respectively. It appears that word level similarity increases the similarity between paragraphs in embeddings as well.

6 Task 3: Statute Law Information Retrieval

The statue law information retrieval task focuses on finding relevant articles $S_1, S_2, ..S_N$ from the complete Japanese Civil Code to a legal bar exam question Q, such that the use of these articles in combination with the exam question would yield entailment or not. We use the English version of this data, which is the translated version of the original Japanese dataset.

6.1 Methodology

We will use three different approaches to generate a feature embedding for each bar exam question and all articles. These are then compared and ranked using cosine similarity. We then always return the most similar article, and any additional ones based on the difference δ between the previously retrieved article and the next most similar article. If that difference is below a certain threshold H, we continue to return more articles until either the threshold is exceeded or a maximum amount of retrieved articles R is found. To evaluate our models, we use the macro-average of precision, recall and the F-2 measure, where the latter is chosen due to the higher importance of recall.

Approach 1 - TF-IDF Vectors This feature approach, which is based on last year's winner [15], is TF-IDF with IDF weight smoothing. We generate TF-IDF vectors for each article and bar exam question, containing the relative frequency of each word in a given vocabulary.

Approach 2 - Legal Embeddings The other approach is based on our legal embeddings, discussed in Section 3. Embeddings have been proven to capture context, whereas a simple word approach as TF-IDF does not, hence the choice of using embeddings as well. For all the articles and bar exam questions, we generate a legal question/article embedding by finding the legal embeddings for its words and take the average for the final representation.

Approach 3 - Combination Given that the earlier mentioned approaches have their own shortcomings as well as strengths, we propose another approach to combine the best of both. We use both approaches separately and combine their top 100 retrieved articles and re-evaluate the order with the same ranking mechanism as explained before.

6.2 Experimental Setup

Since each of these approaches has parameters to tune, we use a grid search method to find the best parameters via the hold out validation set. This yields for the TF-IDF implementation a maximum amount of 3 retrieved articles and a $\delta = 0.007$ which results in a larger recall over precision. For the legal embedding approach we set a maximum amount of 4 retrieved articles and $\delta = 0.007$. Lastly, for the combined model, the maximum amount of retrieved articles is 4 and $\delta = 0.015$. Moreover, the embeddings use lowercasing, punctuation removal, stopword removal and number to text conversion. For the TF-IDF, we keep casing, do not remove punctuation, use number to text conversion, keep stopwords and use both uni- and bigrams.

We split our training data into a train and validation set to obtain intermediate results as well as having interpretable numbers during hyper-tuning and use 600 examples for training and 96 for validation. These results are shown in Table 3, where we can observe that recall is indeed larger. Critically, the combination of the two features show a more balanced precision and a larger recall.

Approach	Precision	Recall	F2
TF-IDF	0.5130	0.5217	0.5117
LE	0.3823	0.5382	0.4490
Combination	0.4790	0.5660	0.5161

Table 3. Validation results of the models for task 3, where LE stands for the legal embedding approach.

6.3 Results

When evaluating our models against other COLIEE 2020 participants on the test set, we get the results mentioned in Table 4. Our legal embeddings on their own do not cover enough semantics and context to obtain sufficient performance; however, in combination with TF-IDF they do boost the recall significantly, showing that they have promising prospects.

Team	Precision	Recall	F2	MAP	R@5	R@10	R@30
LLNTU	0.6875	0.6622	0.6587	0.7604	0.8071	0.8571	0.9214
JNLP.tfidf-bert-ensemble	0.5766	0.5670	0.5532	0.6618	0.6857	0.7143	0.7786
cyber1	0.5058	0.5536	0.5290	0.5540	0.5500	0.6929	0.8000
TAXI_R3	0.4393	0.5089	0.4546	0.5057	0.5714	0.6143	0.6786
TAXI_R1	0.4435	0.4152	0.4112	0.4883	0.5857	0.6214	0.7214
TAXI_R2	0.2872	0.4182	0.3400	0.3741	0.3786	0.4214	0.5643

Table 4. Quantitative results on task 3 by the top-3 participants on COLIEE 2020 and our three submissions. R1, R2 and R3 stand for the TF-IDF, Legal Embedding and combination approach respectively.

Analysis

Training Statistics When we evaluate the cosine similarities values between the bar exam questions and all possible articles, we can see a clear difference in their similarity values, which is shown in Table 5. The TF-IDF similarity values indicate a large difference between relevant and non-relevant articles compared to a bar exam question. However, there is a large standard deviation among relevant articles, indicating that some relevant articles to bar exam questions are not found by TF-IDF which are found by our legal embeddings.

Approach	Relevant	Non-relevant
TF-IDF	0.1651 ± 0.1629	0.0092 ± 0.0184
LE	0.9264 ± 0.0436	0.8754 ± 0.0437

Table 5. Mean cosine similarity value ± their standard deviation for relevant and non-relevant articles to the bar exam questions on the training data with both feature methods.

Strengths & Shortcomings When we manually evaluate the performance of both the TF-IDF and legal embeddings on a few test examples, we can see a clear difference in their strengths and shortcomings. TF-IDF tends to work quite well on finding relevant articles to bar exam questions when the vocabulary used is the same. However, our legal embeddings also find the correct relevant articles to a bar exam question even without shared vocabulary between them. For example, it has learned that a *higher statutory agent* can also be a *legal representative*, which gives us a good indication that these embeddings are essential for improved text comparison.

7 Task 4: Statute Law Entailment

The statue law entailment task requires determining whether a legal bar exam question, Q, is entailed in the text of a set of articles, $S_1, S_2, ...S_N$ relevant to Q. Entailment is taken to mean whether Q is true or false given the content of $S_1, S_2, ...S_N$. This was done in two ways: using a BERT-XGBoost combination and using the legal embeddings with a Bi-GRU. The former is inspired by the previous 2019 COLIEE submission of Gain and colleagues (2019) for the purposes of bench-marking the latter.

7.1 Methodology

BERT-XGBoost Combination For each example, the set of articles S were concatenated to each other, then to the question, Q, with a separator token, and summarized with Gensim if necessary. The last hidden layers of the BERT base cased model were then input to XGBoost.

Legal Embeddings Bi-GRU For each example, the set of articles S were concatenated to each other and then to the question, Q, with a separator token. Stop words were not removed as they contain important information regarding negation and affirmation. The tokenized and padded data was then fed to the Bi-GRU.

7.2 Experimental Setup

XGBoost, was then hyperparameter tuned using five-fold cross validation and Bayesian optimisation methods. The hyperparameters tuned were the following: subsample; the amount of training data to be used per sample, max depth; the maximum tree depth, eta; the learning rate, colsample by level; the subsample ratio per tree used when making a level, and colsample by tree; the subsample ratio per column used when making a tree. Resulting values are .60, 4, .50, .34, .98, respectively. The tuned model was evaluated on a hold out validation set of 20% of the data yielding an average precision, recall and F1 of .63.

Initial experiments found that the Bi-GRU-ATT used in previous legal prediction tasks [3] has too many parameters for such a small dataset. Therefore, we used only one GRU layer and removed the attention. Due to time constraints the model was not fully hyperparameter tuned. The resulting precision, recall, accuracy and F1 on the test set were .59, .58, .58, .56, respectively.

7.3 Results

The results of our two submissions, taxi_BERTXGB and taxi_le_brigru, can be found in Table 6 alongside the top three wining submissions. The former answered 60 questions correct with an accuracy of 0.5357 and the latter answered 57 questions correct with an accuracy of 0.5089.

Model	Number Correct	Accuracy
JNLP.BERTLaw	81	0.7232
TRC3mt	70	0.6250
TRC3t5	70	0.6250
taxi_BERTXGB	60	0.5357
taxi_le_bigru	57	0.5089

Table 6. The top three performing models of COLIEE 2020 task 4 along with our two submissions.

Analysis By using TF-IDF cosine similarity as a measure of shared vocabulary we can elucidate both models' strengths and shortcomings.

The necessity of shared vocabulary for the BERTXGB implementation is evident in the similarity difference between correctly and incorrectly predicted labels, 0.4252 and 0.3429, respectively. However, BERTXGB was not competitive with the top performing submissions. This is likely due to the fact that BERT cased has not been trained on legal text.

The shortcomings of the Bi-GRU model are less explicit, as the TF-IDF cosine similarity is actually higher in the incorrectly predicted labels than in the correctly predicted labels. Indeed, the Bi-GRU implementation was not much better than random chance.

8 Conclusion

In this paper, we propose several approaches to both legal information retrieval and entailment. For task 1, the case law retrieval task, we use CatBoost with both absolute word counts and TF-IDF features obtaining a F1 score of 0.04. The second task, the case law entailment task, makes use of Information Retrieval features such as BM-25, NLI probabilities and fine-tuned embeddings. This model achieves a F1 score of 0.6180, while also achieving third place in the competition. The third task, the statutory information retrieval task, utilizes pre-trained legal embeddings in combination with TF-IDF obtaining a F2 of 0.4546. Lastly, task 4, the statutory entailment task, achieves an accuracy of 0.5357 with BERT embeddings into XGBoost.

The use of extra linguistic features, such as NLI, have been shown to be important and useful to obtain better and state-of-the-art performance. Moreover, we showed that even though the current legal embeddings are not state-of-the-art, they do indicate an understanding of legal terms that is necessary for obtaining better performance.

Both this and earlier years do show that on average scores on the Japanese data is higher than the English one. Since both languages differ much in semantic and structural properties, it would be worth checking whether the Japanese text contains richer token information to expand to multilingual models. Operationally, we could have better shared methodologies across similar tasks. Sharing of best practices could have no only prevented the human error, but also lead to improved performance across the board.

Future research will involve re-training the legal embeddings using a contextually sensitive model such as BERT for a deeper understanding of legal nuance, and focusing on a more precise low complexity model to transform the rich word legal embeddings to rich legal document representations. Furthermore, we see future opportunities in the addition of new tasks to the competition given our perspective as one of the few participants from the private sector. First, there is industry demand for argumentation mining systems capable of performing the following sub-tasks: extraction from unstructured text, type classification, and relation identification. There is currently only one dataset containing annotated legal argumentation structures [30]. Second, there is industry demand for functionality to score the complexity of legal documents to estimate the difficulty of taking on a case. The European Court of Human Rights ascribes an importance level to each case in the meta data which can be used as a proxy for a complexity label [21].

References

1. Benjamins, V.R., Casanovas, P., Breuker, J., Gangemi, A.: Law and the semantic web: legal ontologies, methodologies, legal information retrieval, and applications, vol. 3369. Springer (2005)
2. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
3. Chalkidis, I., Androutsopoulos, I., Aletras, N.: Neural legal judgment prediction in english. arXiv preprint arXiv:1906.02059 (2019)
4. Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I.: Extreme multi-label legal text classification: a case study in eu legislation. arXiv preprint arXiv:1905.10892 (2019)
5. Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y.: Xgboost: extreme gradient boosting. R package version 0.4-2 pp. 1–4 (2015)
6. Clement, J.: Worldwide desktop market share of leading search engines from january 2010 to april 2019. Retrieved March **12**, 2019 (2019)
7. Dale, R.: Law and word order: Nlp in legal tech. Natural Language Engineering **25**(1), 211–217 (2019)

8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
9. Dorogush, A.V., Ershov, V., Gulin, A.: Catboost: gradient boosting with categorical features support. arXiv preprint arXiv:1810.11363 (2018)
10. Gain, B., Bandyopadhyay, D., Saikh, T., Ekbal, A.: Iitp in coliee@ icail 2019: Legal information retrieval using bm25 and bert (2019)
11. Holzenberger, N., Blair-Stanek, A., Van Durme, B.: A dataset for statutory reasoning in tax law entailment and question answering. arXiv preprint arXiv:2005.05257 (2020)
12. Introna, L.D., Nissenbaum, H.: Shaping the web: Why the politics of search engines matters. The information society **16**(3), 169–185 (2000)
13. Kerikmäe, T., Hoffmann, T., Chochia, A.: Legal technology for law firms: determining roadmaps for innovation. Croatian International Relations Review **24**(81), 91–112 (2018)
14. Kim, M.Y., Rabelo, J., Goebel, R.: Statute law information retrieval and entailment. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. pp. 283–289 (2019)
15. Kim, M.Y., Rabelo, J., Goebel, R.: Statute law information retrieval and entailment. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law. p. 283–289. ICAIL '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3322640.3326742, https://doi.org/10.1145/3322640.3326742
16. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International conference on machine learning. pp. 1188–1196 (2014)
17. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach (2019)
18. Manning, C.D., Schütze, H., Raghavan, P.: Introduction to information retrieval. Cambridge university press (2008)
19. Mitra, B., Craswell, N.: Neural models for information retrieval. arXiv preprint arXiv:1705.01509 (2017)
20. Nielsen, D.: Tree boosting with xgboost-why does xgboost win" every" machine learning competition? Master's thesis, NTNU (2016)
21. OPIJNEN, M.V.: Towards a global importance indicator for court decisions. In: Legal Knowledge and Information Systems: JURIX 2016: The Twenty-Ninth Annual Conference. vol. 294, p. 155. IOS Press (2016)
22. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: Catboost: unbiased boosting with categorical features (2017)
23. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M., et al.: Okapi at trec-3. Nist Special Publication Sp **109**, 109 (1995)
24. Rocktäschel, T., Grefenstette, E., Hermann, K.M., Kočiskỳ, T., Blunsom, P.: Reasoning about entailment with neural attention. arXiv preprint arXiv:1509.06664 (2015)
25. Rossi, J., Kanoulas, E.: Legal information retrieval with generalized language models. Proceedings of the 6th Competition on Legal Information Extraction/Entailment. COLIEE (2019)
26. Saha, P., Mathew, B., Goyal, P., Mukherjee, A.: Hateminers: detecting hate speech against women. arXiv preprint arXiv:1812.06700 (2018)

27. Salehi, S., Du, J.T., Ashman, H.: Use of web search engines and personalisation in information searching for educational purposes. Information Research: An International Electronic Journal **23**(2), n2 (2018)
28. Sammons, M., Vydiswaran, V.V., Roth, D.: Ask not what textual entailment can do for you... In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 1199–1208 (2010)
29. Shelmanov, A., Pisarevskaya, D., Chistova, E., Toldova, S., Kobozeva, M., Smirnov, I.: Towards the data-driven system for rhetorical parsing of russian texts. In: Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019. pp. 82–87 (2019)
30. Teruel, M., Cardellino, C., Cardellino, F., Alemany, L.A., Villata, S.: Increasing argument annotation reproducibility by using inter-annotator agreement to improve guidelines. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (2018)
31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
32. Wehnert, S., Hoque, S.A., Fenske, W., Saake, G.: Threshold-based retrieval and textual entailment detection on legal bar exam questions. arXiv preprint arXiv:1905.13350 (2019)
33. Wydick, R.: Plain english for lawyers: Teacher's manual (2005)
34. Yoshioka, M., Song, Z.: Hukb at coliee 2019 information retrieval task - utilization of metadata for relevant case retrieval. Proceedings of the 6th Competition on Legal Information Extraction/Entailment. (2019)
35. Zhong, H., Xiao, C., Tu, C., Zhang, T., Liu, Z., Sun, M.: How does nlp benefit legal system: A summary of legal artificial intelligence. arXiv preprint arXiv:2004.12158 (2020)
36. Zientek, L.R., Werner, J.M., Campuzano, M.V., Nimon, K.: The use of google scholar for research and research dissemination. New Horizons in Adult Education and Human Resource Development **30**(1), 39–46 (2018)