Detecting Relevant Differences Between Similar Legal Texts

Xiang Li*, **Jiaxun Gao***, **Diana Inkpen,** and **Wolfgang Alschner**
University of Ottawa
{xli355,jgao081,diana.inkpen,wolfgang.alschner}@uottawa.ca

Abstract

Given two similar legal texts, is it useful to be able to focus only on the parts that contain relevant differences. However, because of variation in linguistic structure and terminology, it is not easy to identify true semantic differences. An accurate difference detection model between similar legal texts is therefore in demand, in order to increase the efficiency of legal research and document analysis. In this paper, we automatically label a training dataset of sentence pairs using an existing legal resource of international investment treaties that were already manually annotated with metadata. Then we propose models based on state-of-the-art deep learning techniques for the novel task of detecting relevant differences. In addition to providing solutions for this task, we include models for automatically producing metadata for the treaties that do not have it.

1 Introduction

Legal documents typical use standardized forms and structures ("boilerplate language"). Moreover, within a given domain, legal documents often follow model texts and templates resulting in shared norms, principles and dispute resolution mechanisms. However, faced with high-similarity texts, what matters most to lawyers are often textual differences. Where does a contract deviate from an industry standard? How does a law differ from an international model law? And when are these differences legally relevant rather than just stylistic?

Our work seeks to detect such relevant differences between otherwise similar legal texts. It uses international investment treaties as a case study. Table 1 and table 2 provide examples to show what we mean by "relevant" differences. Both of these two sentence pairs have cosine similarity scores around 0.97 when applying LegalBERT (Chalkidis et al., 2020a) to represent them as dense vectors.

Sentence 1
The right of each contracting party to establish its own domestic labour standards and to adopt or modify accordingly its labour legislation each contracting party shall strive to ensure that its legislation provide for labour standards consistent with the internationally recognised labour rights set forth in paragraph 6 of article 1 and shall strive to improve those standards in that light.
Sentence 2
Recognising the right of each contracting party to establish its own levels of domestic environmental protection and environmental development policies and priorities and to adopt or modify accordingly its environmental legislation each contracting party shall strive to ensure that its legislation provide for high levels of environmental protection and shall strive to continue to improve this legislation.
Similarity Score: 0.9734

Table 1: Example of *relevant* difference

Sentence 1
Contracting party shall promptly respond to specific questions and provide upon request information to the other contracting party on matters referred to in paragraph 1 of this article.
Sentence 2
Each contracting party shall upon request by the other contracting party promptly respond to specific questions and provide that other contracting party with information on matters set out in paragraph 1.
Similarity Score: 0.9746

Table 2: Example of *stylistic* difference (not semantically relevant)

However, the sentences in table 1 refer to different subjects even though they share a very similar structure: one deals with labour standards; the other talks about environmental protection. This is an example of a relevant difference which would catch the attention of legal researchers. On the contrary, the sentences in table 2 are similar representations with the same legal meaning and are thus not of interest to legal researchers; we call them stylistic differences. Sentences in table 3 differ completely in semantics and structure. However, due to their highly overlapping vocabulary, they would be extracted as similar sentences. The examples are articles from the Electronic Database of Investment

*These authors contributed equally to this work.

Proceedings of the Natural Legal Language Processing Workshop 2022, pages 256 - 264

Sentence 1
Case of reinvestment of returns from the investments these reinvestments and their returns will enjoy the same protection as the initial investments.
Sentence 2
Each contracting party shall accord at all times fair and equitable treatment to investments of investors of the other contracting party.
Similarity Score: 0.8416

Table 3: Example of *irrelevant* difference (not relevant in sentence structure)

Treaties (EDIT) (Alschner et al., 2020), a resource that we will use in this work, as described later, in section 4.

Traditional measures, such as cosine similarity between TF-IDF (term frequency / inverse document frequency) vectors to represent sentences, fail to capture semantic information crucial for separating stylistic and semantic similarity. The variety of the expressions in these texts can easily mislead word-based approaches to provide similarity scores that are too low. At the same time, small but relevant differences can be easily overlooked if state-of-the-art sentence embedding models are applied directly.

In this paper, we address these challenges by proposing a text difference detection model which is trained on international legal treaties to indicate relevant differences between otherwise similar articles.

2 Related Work

There is a growing body of research on Natural Language Processing and Machine Learning techniques for legal applications. The applications that focus on legal text processing can be divided by the type of text: court judgements and related types of texts on one side, and contracts, treaties, or statutes on the other side.

The tasks addressed vary, from information retrieval from large amounts of legal text, to legal text summarization, legal named entity extraction, court judgement prediction, and more. Pre-trained neural language models were developed for English texts, such as LegalBERT (Chalkidis et al., 2020a), as well as for a few other languages (Masala et al., 2021) (Douka et al., 2021).

Common shared legal text mining tasks are exemplified by SemEval-2023 Task 6 LegalEval: Understanding Legal Text [1] which has three subtasks: predicting the rhetorical roles of sentences (such as preamble, fact, ratio, arguments, etc.), legal named entity extraction, and court judgement prediction with explanation. Similarly, the Artificial Intelligence for Legal Assistance (AILA 2021) shared task at FIRE 2021 [2] included a rhetorical role labelling task continued from previous editions, and legal judgement summarization task. (Parikh et al., 2021). Finally, the Competition on Legal Information Extraction/Entailment (COLIEE 2022) [3] included tasks relating to case law and statutory law such as a legal case retrieval task, a legal case entailment task, a Question Answering system based on relevant statutes from a database of Japanese civil code statutes and entailment of a yes/no answer from the retrieved civil code statutes. The solutions used to solve these tasks involved classical information retrieval methods, while a few applied deep learning methods for retrieval (Rabelo et al., 2022).

Specifically relating to statutory law type documents, such as contracts, laws and treaties, there is a growing interest to automatically identify similarities between documents. Use-cases include identifying where national laws implement international laws (Nanda et al., 2019). In addition, researchers have attempted to assess to what extent legal texts copied from each other or from model agreements (Ash and Marian, 2019) (Allee and Elsig, 2019).

While these studies provide insights into document similarity, most legal scholars are interested not in how similar documents are but where and how similar documents differ, as discussed in (Alschner, 2018). Standard text difference detection algorithms (such as diff in linux/unix) are not able to detect which differences are relevant from a semantic point of view and which are not.

Therefore, our task is different from the tasks addressed in related work or in the shared tasks. We are also using a dataset of legal texts that has not been exploited before by computational methods.

3 Definitions

3.1 Document Hierarchy and Structure categories of Articles

A treaty is a highly standardized legal document. It is composed of articles that divide the treaty into "structure categories" such as Preamble, Defini-

[1] https://sites.google.com/view/legaleval/
[2] https://sites.google.com/view/aila-2021
[3] https://sites.ualberta.ca/ rabelo/COLIEE2022/

tions, Exceptions or Final Provisions. Within each of these structure categories, articles can be further classified according to their content. We call these subcategories "content categories". Each article can have multiple content categories but can only belong to exactly one structure category. The structure categories and the content categories together form a tree-like hierarchy of article categories.

3.2 Keyword Mapping and Content Categories of Articles

In the EDIT database, articles were manually classified into structure categories. Keywords were then used to map articles to different content categories such as Sustainable Development, Governance, or Environmental Protection. Articles can match with multiple keywords. In that case, all corresponding content categories are assigned to an article.

3.3 Relevant Difference

A relevant difference is an abstract concept that is not the same as differences that are very obvious or too trivial. A relevant difference should be more substantial than a simple replacement of synonymous words (a "stylistic difference"), but less than a difference in structure categories (involving unrelated clauses). For the purpose of this project, sentences within the same structure category but within different content categories are considered relevant differences.

4 Data

The data used in the paper origins from the Electronic Database of Investment Treaties (EDIT) (Alschner et al., 2020), which is a new comprehensive full-text database of international investment agreements (IIAs). It contains 3,786 international treaties. In EDIT, all articles with an article title have their structure category labeled through a manual assignment by experts. 71 different structure categories exist in the dataset. In addition, articles in the treaties were further classified into 144 content categories according to 702 different keywords.

For the task we address in this paper, we need sentence pairs with high similarity to be classified into exhibiting a relevant, stylistic, or irrelevant difference. Instead of asking human judges to label pairs of texts, we use the existing EDIT metadata to construct the labels we need for our training and test data. We first extracted sentence pairs with high similarity scores evenly from within each structure category. We then used the keyword-category mapping from figure 1 to automatically label a dataset of sentence pairs for our task. For each two highly similar sentences (with similarity score bigger than 0.965 and less than 0.98)[4], we label them as only displaying a stylistic difference if both of them share a same set of content categories (according to the keywords they contain). In contrast, if the proportion of overlapping categories is less than one third of the sentence pair's content category union, we consider that one sentence discusses a subject or topic that is distinct from the other and that these two sentences thus have relevant differences. Moreover, we also introduced less similar sentences (with a score less than 0.85)[5] as examples of sentences having irrelevant differences. Via this method, we obtained a dataset with 12,968 sentence (article) pairs. 8,430 of them are sentences that have the same content categories and therefore are labelled as having only stylistic differences (no semantic or legal differences). 2,096 of the sentence pairs are labelled as containing relevant differences which would be the ones of interest to a legal researcher. We also introduced 2,442 sentence pairs which are less similar from each other and labeled as having irrelevant differences. This is done to better simulate common application scenarios[6]. We use this dataset of sentence pairs to train our automatic methods for detecting relevant differences. First, we keep aside 20% of the dataset

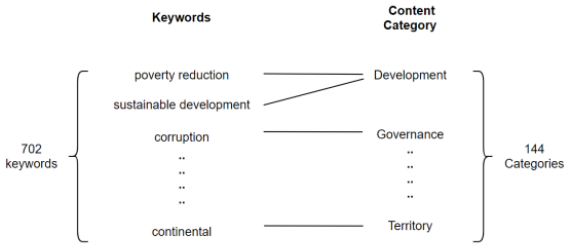

Figure 1: Example of keyword mapping

[4]Note that these values were chosen in order to produce candidate pairs; they do not affect the labels that will be assigned to them.

[5]This value is selected by observation of the experimental results.

[6]In real-world situations, this kind of irrelevant difference appears pretty commonly when trying to identify similar sentences. We incorporated this irrelevant data in the training process so that the model can better identify them and the final accuracy.

for testing the models that we will train. Figure 2 shows the distribution of the three classes in our dataset.

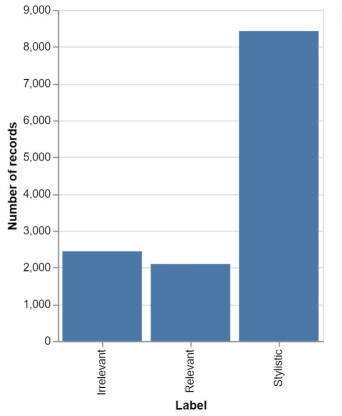

Figure 2: Distribution of labels in the constructed dataset for relevant differences detection

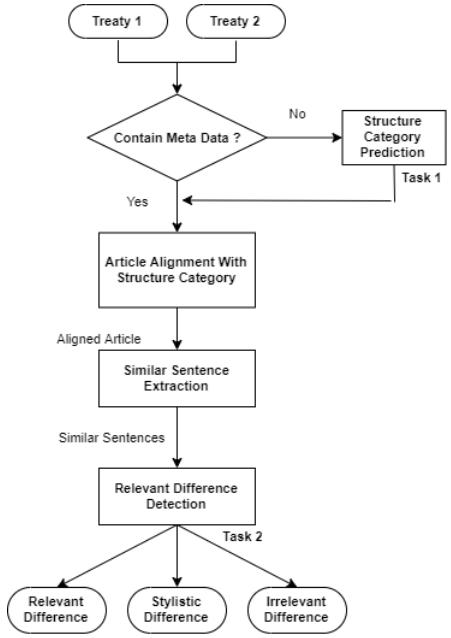

Figure 3: Workflow of relevant difference detection

5 Tasks

The main task of this paper is to distinguish relevant differences from stylistic or irrelevant differences between similar texts, in order to facilitate legal research. This means to ignore differences that are too small and uninteresting from a semantic point of view. At the same time, sentences that are very different are not of interest since they are easy to identify (such sentences were not included in our dataset). To achieve our goal, a few preprocessing tasks (explained below) were be performed in order to build a dataset of similar sentences labeled for relevant differences, to allow us to train the models and evaluate them. Figure 3 shows our workflow.

5.1 Structure Category Prediction

Given two treaties, the first step is to verify whether they contain article meta data. This meta data was manually assigned and is used to match similar articles. As mentioned in section 4, EDIT contains labels for the structure categories for most of the articles. However, there are still 1,052 articles without any meta data. These articles do not contain article title texts and could therefore not be categorized by the experts. As a result, not all the treaties contain structure categories for articles. In this circumstance, an additional classification of articles based on their structure category is required for the unlabeled articles before further analysis on relevant differences can be conducted. As a secondary task, we thus assess the feasibility of assigning structure categories automatically. This will be especially useful when new treaties will be added to EDIT, to avoid the need for more manual annotation.

5.2 Detecting Relevant Differences

After the topic classification, all articles are now labeled with structure categories. An alignment can be constructed between articles that share the same structure category. Similar sentences (having similarity score larger than 0.9) from the aligned articles are extracted and send to further automatic processing for the relevant differences detection. As mentioned above, our models will predict one of three classes: stylistic difference, relevant difference, and irrelevant difference.

6 Methods

6.1 Methods for Structure Category Prediction

6.1.1 Dataset Preprocessing

From all 3,786 legal treaties in EDIT, we extracted 27,530 articles having structure category label as training dataset for topic prediction and 6,883 articles as a separate test dataset. These articles are uniquely labelled with 71 different structure categories (manually entered in EDIT, as mentioned). Inspection on category distribution shows that the training dataset is highly imbalanced. Therefore, we replaced all categories which contain fewer than

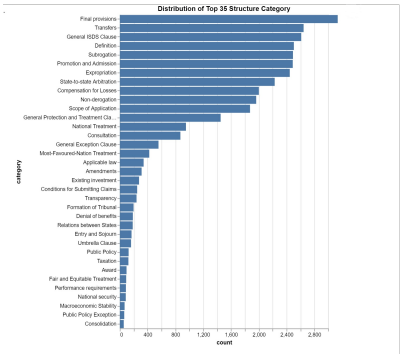

Figure 4: Structure categories Distribution (top 35)

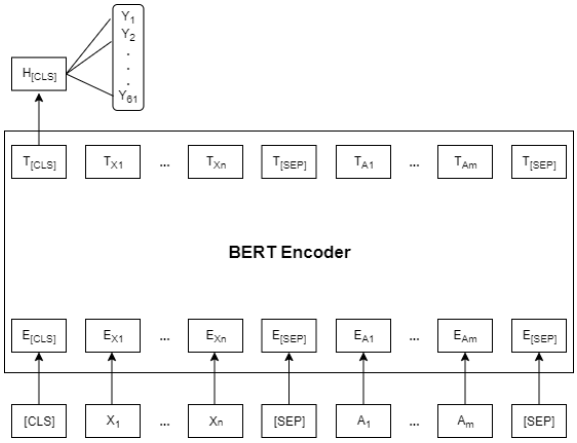

Figure 5: Structure of the proposed model for topic classification

5 articles with the label "other". This is applied to reduce the number of categories and to avoid overfitting. After the category replacement, 61 structure categories remained. See Figure 4 for examples of the most frequent categories. We also applied pre-processing steps such as lower-case conversion, stop words removal and lemmatization before further exploration.

6.1.2 Models

Baseline Models: To provide a point of reference for advanced models based on deep learning, we firstly trained a SVM model for the structure category prediction, as a baseline. A 100-dimension TF-IDF vectorization was applied on the corpus after preprocessing pipeline. We finally derived a 27,530 $\times$ 100 sparse matrix with 582844 non-zero elements as feature space and trained a linear-kernel SVM on it. For another model, we employed averaged word vectors (word2vec pretrained on the Google News corpus, with 300 dimensions) over the words composing the sentence as features. For this dense feature space, we applied an RBF-kernel SVM as the classifier.

BERT-based Models: For a state-of-the-art model for our task of topic classification, we designed a BERT-based model (Devlin et al., 2019) to predict the structure category from existing labeled articles via constructing auxiliary sentences and incorporating context knowledge (as explained below).

Our proposed model consists of three parts, also illustrated in figure 5:

1. The first input layer part aims to construct input sequence from given data.The WordPiece tokenization is applied to convert the input article into tokens and adds the [CLS] and [SEP] token as separator. The position embeddings, word embeddings and segmentation embeddings for each token are then summed up to yield the final input representations.

2. The second BERT encoder part consists of 12 Transformer blocks and 12 self-attention heads by taking as input a sequence and outputting its representations.

3. The third output layer is composed by a simple softmax classifier taking the input from vector embedding of token [CLS].

For this task, inspired by the standard structure of legal treaties, we set the input sequence of our model as a combination of the article to be predicted and its succeeding article in the original treaty, to provide context. The article having the last position in a treaty will be transmitted twice if it is chosen as input. To allow comparisons with other BERT-based classification models whose input only consists of a simple text sequence, we experimented with the construction of input in both ways, with two models, as illustrated in figure 6:

- **BERT-base-S** for single input sequence with article to be predicted.
- **BERT-base-A** for input sequence contains article to be predicted along with auxiliary succeeding article.

Considering the length of articles, we set the input size to 512. Deducting the 3 tokens occupied by [CLS] and [SEP], only at most 509 tokens are reserved for input articles. When the sum of the target article of size n and the auxiliary article of size m exceeds 509, we choose to keep the article

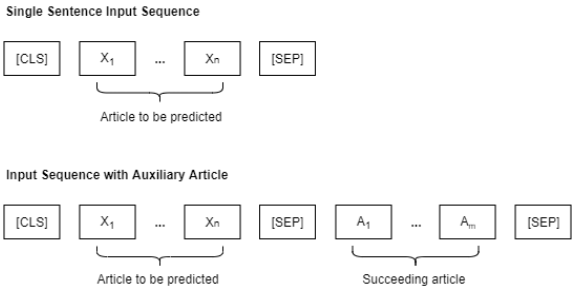

Figure 6: Two ways for input construction

to be predicted and only shorten succeeding the auxiliary article to size $(509 - n)$.

The output layer is a softmax classifier on the top of BERT encoder which maps the 768-dimension vector $H_{[CLS]}$ into the conditional probability distributions:

$$P(y_i|H_{[CLS]}, \theta) = softmax(W^T H_{[CLS]})$$

$$= \frac{exp(W^T H_{[CLS]}[i])}{\sum_{j=1}^{61} exp(W^T H_{[CLS]}[j])} \quad (1)$$

over all labels $y = \{y_1, y_2, ..., y_{61}\}$ where θ is the set of all trainable parameters and $W \in \mathbf{R}^{768\times61}$ is the weight matrix of the classifier.

We take $\hat{y} = argmax(P(y_i|H_{[CLS]}, \theta))$ as the predicted result and calculate the loss based on the cross-entropy function.

6.2 Methods for Relevant Difference Detection

6.2.1 Data Preprocessing

During the experiments, we found that BERT models, especially LegalBERT, are very sensitive to minor changes in articles. Even different notations appearing in sentences will lead to a lower similarity score and label two identical articles as different. Therefore, before further exploration, we first cleaned the dataset and removed misleading notations such as indices before treaties.

Another important step in our data preprocessing pipeline is categorical keyword removal. We replaced 85% tokens which are contained in the category keyword list with <MASK>. This procedure is motivated by the following observations: the correlation between the keyword contained in the sentence and the article category is high; this indicates that if we keep the keyword in the dataset, the model will be very likely to overfit. Moreover, if the categorical keywords have not been filtered out, the model will focus on the existing keywords and lose the generality to perform well on unseen categories. The threshold value 85% was chosen empirically and was inspired by BERT pre-training.

Dataset	Train Acc	Test Acc
keywords 85% removed	0.87	0.85
keywords 100% removed	0.74	0.71
keywords all kept	0.91	0.78

Table 4: Keyword removed vs Keyword kept

To demonstrate the above hypothesis, we experimented with three datasets, one with all keywords being kept, one with 85% keywords removed, and another one with all keywords being removed. We trained a CNN model for 10 epochs with FastText embeddings on the three datasets and obtained different results. Table 4 shows that keeping all keywords in the sentences can harm the generality of the network. Removing all the keywords will lead the model toward underfitting. As a result, keeping 15% of the categorical keywords achieved the best results among the three datasets. Therefore, we use this dataset in the following experiments.

The following is the summary of the preprocessing steps that we used for this task:

- Converting to lower case
- Remove indices before treaties
- Remove stop words and punctuation marks
- Convert word numbers to numeric form
- Correct wrong spelling
- Remove the identified category keyword on a small portion of training data

In the above procedure, the FastText library was used for word embeddings, word2number[7] was used for the conversion from word numbers to numeric value and a spell checker was applied on the dataset to correct all typos.

6.2.2 Models

Before the modeling, a train-test split was performed. We trained all our models on 80% of the data and the other 20% of the data was left for testing purposes. We used accuracy, precision, recall, and the F1-score as evaluation metrics to assess the performance of each classifier. We evaluated two

[7] https://pypi.org/project/word2number/

classical machine learning models and five deep learning methods, including three BERT-based classifiers, to detect relevant differences based on sentence pairs.

In all the models described below, we combined two sentences by a <SEP> token and fed the concatenated tokens to the model.

Baseline Methods: We used Mutinomial Naive Bayes and XGBoost decision tree as baseline classifiers. We included the document length, word counts, and n-gram TF-IDF representations as statistical features. The performance of the above classical approaches will be reported in section 7.

Deep Learning Approaches:

- **CNN_FastText**: A convolutional model with pretrained embedding was set up for the deep learning baseline. We used the 300-dimensional embedding layer provided by FastText[8] as sentence representation.
- **BiLSTM_FastText**: A bidirectional Long Short-Term Memory (LSTM) model was trained and evaluated on the dataset. Due to the nature of our task, the whole article is required before the inference, so we applied BiLSTM to incorporate the context from both directions.
- **BERT (bert-base)**: We also fine-tuned and evaluated the BERT-base model using pretrained transformer embedding layers provided by Huggingface [9]. To fine-tune the pretrained BERT model for classification, we applied dropout on the <CLS> token and the token was fed to a softmax function. We selected batchsize = 16, learning rate = 1e-6 and dropout = 0.5.
- **legalBERT**: LegalBERT (Chalkidis et al., 2020b) is a version of the BERT-base model that has been specifically trained on legal documents. The embedding representation was trained on 12 GB of diverse English legal text from several fields (e.g., legislation, court cases, contracts). The model was designed to be able to classify legal documents and to extract information from them.
- **RoBERTa**: RoBERTa (Liu et al., 2019) is a highly optimized version of BERT. The pretrained model from Huggingface [10] was fine-tuned on our dataset. The performance comparison between RoBERTa and other transformer-based models will be presented in the next section.

[8]https://fasttext.cc/docs/en/english-vectors.html
[9]https://huggingface.co/bert-base-uncased

7 Results and Discussion

7.1 Results for Structure category Prediction

Model	Prec.	Recall	F1	Acc.
NB_TF-IDF	0.912	0.825	0.849	0.809
SVM_TF-IDF	0.927	0.911	0.917	0.911
SVM_W2V	0.941	0.915	0.927	0.920
BERT-base-S	0.971	0.944	0.957	0.955
BERT-base-A	**0.974**	**0.953**	**0.963**	**0.962**

Table 5: Evaluation results of structure category prediction on the articles from the test data.

Table 5 shows the results on the test data described in section 6.1.1, for two baseline text classification models and for two BERT based models. The best results (marked in bold font) are achieved by our enhanced context-dependent model. These experimental results support our idea that context knowledge provided by the succeeding article helps the prediction of structure category. Another notable fact is that, in this task, all experimented models have precision score higher than recall. This is because the prediction of structure categories is actually a classification with 61 labels. Labels with few articles are less often predicted and hence have lower recall.

7.2 Results for Relevant Difference Detection

Model	Prec.	Recall	F1	Acc.
Multinomial NB	0.621	0.705	0.594	0.592
XGBoost	0.744	0.761	0.754	0.734
CNN_FastText	0.843	0.867	0.858	0.875
BiLSTM_FastText	0.826	0.845	0.835	0.837
BERTbase	0.885	0.911	0.886	0.938
legalBERT	0.864	0.904	0.868	0.913
RoBERTa	**0.940**	**0.956**	**0.939**	**0.960**

Table 6: Evaluation results of different classifiers on the pairs of articles from our test data

Table 6 shows the results on the test data described in section 6.2.1, for two classical machine learning algorithms, as baselines, and the performance of five deep learning algorithms. We can see

[10]https://huggingface.co/roberta-base

that the RoBERTa model achieved the best performance (marked in bold font) among all classifiers. The LegalBERT model is lagging behind despite of being a domain-specific model. The limited performance of LegalBERT was noted in related work (Geng et al., 2021).

We conducted a comprehensive error analysis on the RoBERTa model's output. Among all sentence pairs that have been misclassified, 60% of them are stylistic differences falsely predicted as relevant differences.

Sentence 1
Contracting party shall encourage investments made in its territory by investors of the other contracting party and shall accept such investments in accordance with its laws and regulations.
Sentence 2
Each contracting party shall in its territory promote investments by investors of the other contracting party and admit such investments in accordance with its laws and regulations.

Table 7: Example of stylistic difference misclassified as relevant difference

Table 7 shows such a typical example. Both sentences are in the structure category of "Admission" with minor differences, but they have been classified as having a relevant difference. The possible cause of this misclassification is that the term "encourage" (in the first sentence) might be treated as a keyword mapping to different content categories, and our embedding representation tends to capture this keyword and thus makes our model biased.

To verify this assumption, we performed the same error analysis on the dataset without replacing any keyword as <MASK>, as a result, the proportion of misclassified stylistic difference will increase from 60% to 87%. This increase of false positive rate also verifies the effectiveness of the category keyword removal when constructing the dataset, as mentioned in section 6.2.1.

8 Limitations

We limited our experiments to articles from legal treaties, though our techniques could be applied on any kind of legal texts or even wider, to any similar texts in general language or specific domains. Though a significant barrier on experimenting with other kinds of texts is the lack of annotated data for training supervised classifiers. In our current experiments, we were able to use the already existing manual annotations in EDIT to produce training data of text pairs without the need for new manual work.

Another limitation is caused by the imbalance of the dataset for the structure category prediction. None of existing re-sampling methods seems appropriate to be applied on legal articles, as their structure is highly standardized. Domain-specific re-sampling methods could be further investigated.

9 Conclusion and Future Work

In this paper, we presented several deep leaning based models for the novel task of detecting semantically relevant differences between similar legal texts. In addition, we proposed an enhanced model that uses contextual information for the secondary task of predicting metadata (structure categories). We exploited a valuable legal resource that was not used before for computational analysis of this kind. We are making available our code on GitHub and our datasets with training/test splits for reproducibility purposes[11].

We achieved very good results with the deep learning models that we considered as promising for our tasks, but there are other deep learning models that could be tried in future work.

Another direction of future research is to apply text entailment methods on the articles with relevant differences, to see if one entails the other. This could mean that one treaty was derived from the other one. We could apply this over multiple treaties to trace back the historical evolution of treaty writing. In case the entailment goes in both directions, one article entails the second one and the reverse holds too, this could be another filter to add on top of our best model for detecting relevant differences.

References

Todd L. Allee and Manfred Elsig. 2019. Are the contents of international treaties copied and pasted? evidence from preferential trade agreements. *International Studies Quarterly*.

Wolfgang Alschner. 2018. *Sense and Similarity: Automating Legal Text Comparison*. Edward Elgar.

Wolfgang Alschner, Manfred Elsig, and Rodrigo Polanco. 2020. Introducing the electronic database of investment treaties (edit): The genesis of a new

[11]Code available at: https://github.com/coollx/Relevant-Difference-Detector-for-Legal-Text

database and its use. *World Trade Review*, 20:73 – 94.

Elliott Ash and Omri Y. Marian. 2019. The making of international tax law: Empirical evidence from natural language processing. *InfoSciRN: Natural Language Processing (Sub-Topic)*.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020a. Legal-bert: The muppets straight out of law school. pages 2898–2904.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020b. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Stella Douka, Hadi Abdine, Michalis Vazirgiannis, Rajaa El Hamdani, and David Restrepo Amariles. 2021. JuriBERT: A masked-language model adaptation for French legal text. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 95–101, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sakbo Geng, Rémi Lebret, and Karl Aberer. 2021. Legal transformer models may not always help. *ArXiv*, abs/2109.06862.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. Cite arxiv:1907.11692.

Mihai Masala, Radu Cristian Alexandru Iacob, Ana Sabina Uban, Marina Cidota, Horia Velicu, Traian Rebedea, and Marius Popescu. 2021. jurBERT: A Romanian BERT model for legal judgement prediction. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 86–94, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rohan Nanda, Giovanni Siragusa, Luigi Caro, Guido Boella, Lorenzo Grossio, Marco Gerbaudo, and Francesco Costamagna. 2019. Unsupervised and supervised text similarity systems for automated identification of national implementing measures of european directives. *Artificial Intelligence and Law*, 27(2):199–225.

Vedant Parikh, Upal Bhattacharya, Parth Mehta, Ayan Bandyopadhyay, Paheli Bhattacharya, Kripa Ghosh, Saptarshi Ghosh, Arindam Pal, Arnab Bhattacharya, and Prasenjit Majumder. 2021. Aila 2021: Shared task on artificial intelligence for legal assistance. In *Forum for Information Retrieval Evaluation*, FIRE 2021, page 12–15, New York, NY, USA. Association for Computing Machinery.

Juliano Rabelo, Randy Goebel, mi-young Kim, Yoshinobu Kano, Masaharu Yoshioka, and Ken Satoh. 2022. Overview and discussion of the competition on legal information extraction/entailment (coliee) 2021. *The Review of Socionetwork Strategies*, 16.