Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 1235–1241
Marseille, 11–16 May 2020

LEDGAR: A Large-Scale Multilabel Corpus for Text Classification of Legal Provisions in Contracts

Don Tuggener, Pius von Däniken, Thomas Peetz, Mark Cieliebak
Zurich University of Applied Sciences, Winterthur, Switzerland
Legartis technology AG, Zurich, Switzerland
{tuge, vode, ciel}@zhaw.ch, thomas.peetz@legartis.ai

Abstract

We present LEDGAR, a multilabel corpus of legal provisions in contracts. The corpus was crawled and scraped from the public domain (SEC filings) and is, to the best of our knowledge, the first freely available corpus of its kind. Since the corpus was constructed semi-automatically, we apply and discuss various approaches to noise removal. Due to the rather large labelset of over 12,000 labels annotated in almost 100,000 provisions in over 60,000 contracts, we believe the corpus to be of interest for research in the field of Legal NLP, (large-scale or extreme) text classification, as well as for legal studies. We discuss several methods to sample subcopora from the corpus and implement and evaluate different automatic classification approaches. Finally, we perform transfer experiments to evaluate how well the classifiers perform on contracts stemming from outside the corpus.

Keywords: multilabel text classification, legal nlp, corpus creation

1. Introduction

Legal Natural Language Processing is an emerging field that investigates the application of Natural Language Processing (NLP) techniques to the legal domain. Several dedicated conferences and workshops have emerged in recent years (Aletras et al., 2019; Rehm et al., 2018; Palmirani, 2018; Keppens and Governatori, 2017). As a relatively young discipline, Legal NLP lacks resources in several areas. We aim to address this lack regarding text classification by presenting LEDGAR (Labeled EDGAR[1]), a corpus of labeled provisions in contracts.[2]

Contractual provisions are a primary research target in law studies (Mills, 2019; Knapp et al., 2019; Hershkoff and Kahan, 2018; Fisch, 2018, inter alia), as they are the essential discourse units when drafting, negotiating, validating, or analysing contracts. Each provision in a contract constitutes a legal speech act (Yovel, 2000; Trosborg, 1995; Trosborg, 1991), and the concatenation of the provisions comprises the legal essence of a contract.

Meanwhile, an emerging line of research investigates text classification with large-scale or extreme labelsets, where labelsets consist of thousands or even millions of labels (Bengio et al., 2019; Choromanska and Kumar Jain, 2019; Soni et al., 2018, e.g.).

In this light, our main contributions are:

- We provide a freely available, substantially-sized corpus (100m+ tokens) that assists the study of provisions, both from a legal and an NLP perspective.
- The corpus features a large labelset (12k+ labels), which makes it attractive to research in the field of large-scale (or extreme) multilabel and multiclass classification.
- Additionally, we discuss several subsampling techniques and present a method for extracting a label hierarchy based on label names. This hierarchy enables us to reduce the initially large labelset to a more feasible size in a standard text classification setting.
- Finally, we demonstrate the corpus' suitability for training classifiers that are applicable to out-of-domain contracts, i.e. contracts that stem from outside the corpus.

2. Related work

We focus on related work in two areas, namely work on extracting corpora from EDGAR or providing structured access to information contained in EDGAR, and research on text classification in the legal domain.

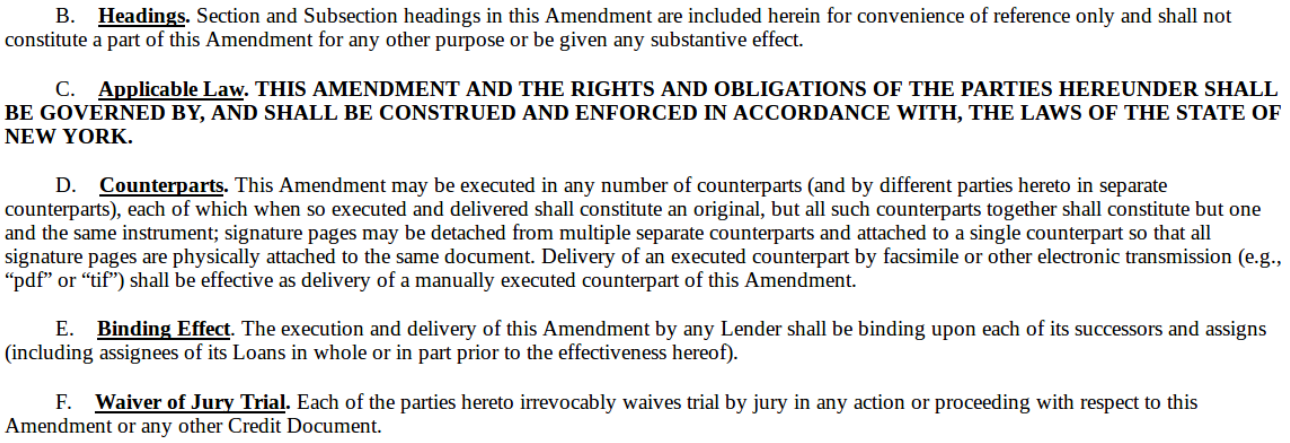
Bommarito et al. (2018b) present an open-source framework to extract and store data from EDGAR in a relational database, effectively making working with EDGAR (e.g. finding Exhibit 10 filings) more convenient. They also include example use cases, such as training word embeddings on press releases contained in EDGAR.

Bommarito et al. (2018a) released an open-source software library for NLP in the legal domain. The library offers support of basic NLP tasks, such as text segmentation, POS tagging, or named entity recognition. Furthermore, it offers heuristics for the extraction and normalization of addresses, amounts, durations, etc. Additionally, the library contains a broad variety of word and document embeddings pre-trained on different legal data (contract types, provision types etc.), and features a binary, machine learning-based classifier to distinguish contracts from other documents.

Chalkidis et al. (2019) investigate large-scale multi-label document classification in the legal domain by predicting concepts occurring in EU legislative documents. They compile a corpus of 57K documents annotated with 4.3K concepts. They find that BERT (Devlin et al., 2018) produces the best classification Micro F1 scores in almost all

[1] Cf. Section 3.1. for the explanation of the acronym.

[2] Available here: `https://drive.switch.ch/index.php/s/j9S0GRMAbGZKa1A`



B. **Headings.** Section and Subsection headings in this Amendment are included herein for convenience of reference only and shall not constitute a part of this Amendment for any other purpose or be given any substantive effect.

C. **Applicable Law. THIS AMENDMENT AND THE RIGHTS AND OBLIGATIONS OF THE PARTIES HEREUNDER SHALL BE GOVERNED BY, AND SHALL BE CONSTRUED AND ENFORCED IN ACCORDANCE WITH, THE LAWS OF THE STATE OF NEW YORK.**

D. **Counterparts.** This Amendment may be executed in any number of counterparts (and by different parties hereto in separate counterparts), each of which when so executed and delivered shall constitute an original, but all such counterparts together shall constitute but one and the same instrument; signature pages may be detached from multiple separate counterparts and attached to a single counterpart so that all signature pages are physically attached to the same document. Delivery of an executed counterpart by facsimile or other electronic transmission (e.g., "pdf" or "tif") shall be effective as delivery of a manually executed counterpart of this Amendment.

E. **Binding Effect**. The execution and delivery of this Amendment by any Lender shall be binding upon each of its successors and assigns (including assignees of its Loans in whole or in part prior to the effectiveness hereof).

F. **Waiver of Jury Trial.** Each of the parties hereto irrevocably waives trial by jury in any action or proceeding with respect to this Amendment or any other Credit Document.

Figure 1: Excerpt from an Exhbit-10 material contract (`https://www.sec.gov/Archives/edgar/data/0001171825/000119312519044328/d691151dex101.htm`) showing different markup of potential labels (underlined and in bold) and the accompanying provisions.

settings in comparison to Logistic Regression and several neural classifiers.

Waltl et al. (2019) explore automatic classification of semantic types of legal norms in German laws. They identify, label, and classify nine functional categories (e.g. *duty, prohibition, permission*) in sentences in the German tenancy law. They transform the texts into TF-IDF representations and found that an SVM classifier performed best in their setting. Waltl et al. (2017) also explore a combination of rule-based classification with active machine learning on this data. Glaser et al. (2018) follow up on this work and explore whether these annotated data are suitable to train classifiers that predict functional categories in rental agreements in a series of experiments similar to ours.

To the best of our knowledge, no prior work is available on provision classification in contracts or on corpora containing provisions labeled in a topical (i.e. not functional) sense.

3. Data

We first describe the data source of our corpus and then outline the extraction of the labeled provisions.

3.1. Data Source

Our corpus is comprised of contracts crawled from the website of the U.S. Securities and Exchange Commission (SEC).[3] The SEC's main mission is to establish transparency of business activities, with the goal of providing investors more security regarding the companies that they invest in. The SEC website hosts a service called EDGAR (Electronic Data Gathering, Analysis, and Retrieval system). Domestic and foreign companies conducting business in the U.S.A. are required to file regular reports to the SEC through EDGAR. Reports are filed based on a list of forms that correspond to certain filing types. While EDGAR features over 150 forms, we targeted filings that contain material contracts (called Exhibit-10), such as agreements (e.g. shareholder/employment/non-disclosure agreements), because a) material contracts include provision types (such as *governing law*) which occur in a broad variety of contracts, and b) these filings offer the opportunity to automatically obtain labels.[4]

3.2. Crawling and Scraping

We crawled all Exhibit-10 contracts from (including) 2016 to 2019, which yields an initial set of 117,578 contracts from which we heuristically scraped labeled provisions. While not all Exhibit-10 filings follow a standardized HTML markup format, we observed regularities in marking up the names of provisions, as shown in Figure 1. That is, provisions are often prepended with the name of their type, and this type is often displayed in special formatting, such as bold-face or underline, followed by a sentence delimiter, such as a dot or colon. We identified two frequent markup strategies for the provision types (i.e. using either `<u>` or `<font>` tags) and heuristically scanned the beginning of paragraphs (identified by `<p>` or `<div>` tags) in contracts for occurrences of the pattern:

```
underlined and/or bold text + delimiter
+ differently formatted text
```

We then treated the specially formatted text as the (potential) label, and the non-formatted text as the (potential) provision text.[5]

To cope with noisy extractions, we applied several filters during the scraping process (e.g. minimum and maximum length of each element in the pattern; first character in text elements must be uppercase; texts cannot consist of stopwords only; labels cannot end in stopwords, etc.).[6] The

[3] `https://www.sec.gov/`

[4] Navigating EDGAR to extract Exhibit-10 filings is a non-trivial task. Our approach is outlined in the codebase that accompanies this paper. A guide to EDGAR can be found under the following URL: `https://www.sec.gov/oiea/Article/edgarguide.html`.

[5] We noted that some provisions feature multiple labels, delimited by a semicolon or a slash. We split such labels into multi-labels.

[6] For the complete set of filters we refer readers to the codebase released with this paper.

	# contracts	# provisions	% with multilabels	# labels
Initial scrape	72,605	1,850,284	5.96 %	183,622
De-duplicate provisions	63,794	1,081,177	6.32 %	182,328
Split labels	63,794	1,081,177	16.69 %	166,063
Merge sg/pl label forms	63,794	1,081,177	16.69 %	158,842
Remove low-frequency labels	61,456	908,465	17.39 %	21,003
Remove outlier labels	60,540	846,274	16.44 %	12,608

Table 1: Impact of provision and label filtering on corpus statistics.

scraping process yielded 1,850,284 labeled provisions in 72,605 contracts and a labelset of size 183,622.

3.3. Data cleanup

We applied several heuristic filters to improve the quality of the extracted data. First, we de-duplicated the provision texts, which reduced the number of provisions from 1,850,284 to 1,081,177. To "sanitize" the labels, we applied the following filters:

Split labels: We found that labels that denote multilabels are sometimes connected by *and* (*Ratification and Acknowledgements*), a comma (*Severability, Modification of Covenants*), or by an ampersand (*Facsimile & Electronic Signatures*). In contrast to semicolons and slashes, we found these delimiters to be ambiguous.[7] If such delimiters occurred, we split the label and verified whether its constituents had been observed as single labels. If so, we assigned the constituents as individual labels, turning the initial label into a multilabel.

Merging singular and plural forms: We merged plural and singular forms of labels by finding all labels ending in *-s* and checking whether a version of the label without the ending *-s* exists in the labelset, e.g. *waiver* → *waivers*. We then merged the singular form label into the plural form.

Pruning labels based on document distribution: Given the relatively large number of documents, we deemed labels idiosyncratic to specific subsets if they occurred in less than five contracts and removed them. We then calculated the distribution of labels over documents with the aim to identify labels that occur in a wide variety of contracts. Our guiding assumption was that labels with a narrow distribution are also idiosyncratic to specific contracts. Clearly, not all labels are similarly frequent. We assumed a linear, monotonic relationship between the frequency of a label and the number of contracts it occurs in. Hence, we applied a linear regression on the frequency of the labels and their document counts to identify outliers. Specifically, for the i-th label, we measured the difference between the predicted label document frequency $\widehat{ldf_i}$ and the observed label document frequency ldf_i, normalized by ldf_i to allow for more variance in frequent labels:

$$dist(ldf_i, \widehat{ldf_i}) = \frac{\widehat{ldf_i} - ldf_i}{ldf_i} \quad (1)$$

Finally, we measured the standard deviation of the distances, and regarded all labels with a negative distance above the standard deviation as outliers. Among the labels deemed outliers by our method are e.g. *New mezzanine loan option*, *Resignation as issuing lender after assignment*, and *Performance of obligations of parent*.

The impact of the provision and label filters on the corpus statistics are shown in Table 1. As a result, 12,608 labels and 846,274 provisions remain after filtering. This is still a comparably large labelset for text classification with a biased label distribution. We show the impact of enforcing minimum count thresholds on the labels on the corpus statistics in Table 2. However, we do not set a threshold for the corpus release, but leave it to practitioners to set their own, or select a subset of labels that they deem relevant. We discuss several subsampling techniques in Section 4.

min. freq.	#contracts	#prov.	#labels	multi
10	60,297	831,283	10,485	16.44%
50	58,437	695,218	2,677	16.41%
100	57,118	625,519	1,442	16.13%
500	53,012	429,771	309	15.01%
1,000	49,922	319,334	130	15.80%
5,000	37,151	123,419	16	14.45%
10,000	30,285	62,701	5	2.88%

Table 2: Corpus statistics (number of labels, provisions, contracts, and percentage of provisions with multilabels) for different label frequency thresholds.

The token and provision statistics of the corpus after the final filtering step are given in Table 3.[8] The corpus contains a substantial amount of tokens suited for e.g. training word embeddings. We also see that the standard deviations for the average of tokens per provision and provisions per contract are quite high, which indicates that there is variety in the corpus.

Quantity	amount (std. dev.)
No. of tokens	104,990,418
Vocabulary size	52,098
Avg. tokens per provision	124 (104)
Avg. provisions per contract	13 (20)

Table 3: Token and provision statistics of the cleaned corpus.

We release the corpus as a `JSONL` file, where each line consists of a `JSON` object that holds the attributes

[7] For example, in the label *Rights of the successor and retiring agent*, *and* is not a connector of two individual labels, while in e.g. *Consideration and Payment* it is.

[8] Note that these statistics only apply to provisions for which we were able to extract labels, other text passages of the contracts are omitted in our scraping.

provision, which contains the provision text, label, which holds the extracted labels, and source, which gives the relative path to the contract from which the provision was extracted.[9] Additionally, we release the codebase used for creating the corpus, as well as for running the classification experiments in the following sections.[10]

3.4. Label hierarchy extraction and label decomposition

A potential means to handle the large labelset in the corpus is to infer a hierarchy of the labels and then predict labels that subsume other labels. We noted that the length of the label names (as measured by number of tokens in the name) seemed to correlate (negatively) with the label frequencies, i.e., labels with longer names tend to be more sparse.

We assumed that the labelset features a latent hierarchy, where longer label names can be considered parents of shorter ones (e.g. *compliance with environmental laws* → *compliance*, *environmental laws*). Hence, we extracted a directed acyclic graph from the labelset, where nodes denote labels, and edges denote subsumption. That is, longer label names point to shorter ones that include consecutive token sequences from the longer label name.

This graph enabled us to decompose (long) label names into multiple shorter labels, i.e. multilabels. Figure 2 shows an excerpt from the label graph, i.e. the label *Adjustment upon subdivision of combination of share of common stock* and how it is decomposed into intermediary and leave nodes. We will explore the use of the graph for labelset manipulation in the next section.

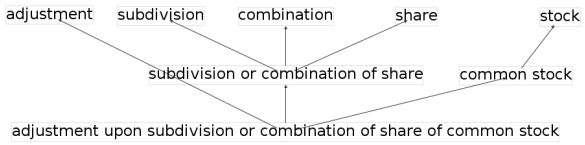


Figure 2: Example of label decomposition.

4. Provision classification

Our corpus of labeled provisions retains all provisions that are left after our cleaning effort. Here, we explore automatic provision classification on four different labelset subsets to verify the suitability of the data for training provision classifiers.

4.1. Data selection

We create the following four data subsets to evaluate automatic provision classification:

Prototypical (*proto*): We calculate the average number of provisions that a contract contains in our corpus, i.e. 13, and select the 13 most common labels based on frequency. This yields 110,156 provisions (12.50% with multilabels) and gives us an approximation of a standard provision inventory in a contract.[11]

Frequency at least 100 (*freq100*): We select all provisions with a frequency of at least 100 in the data. This yields 1,442 labels and 625,519 provisions (16.13% with multilabels).

Projection to leaf labels (*leaves*): We decompose all labels into the leaf labels using the label graph introduced in Section 3.4. The provision count remains unchanged (832,513), while we obtain 945 leaf labels. Here, we obtain a high rate of multilabels due to the decomposition of labels (59.19% with multilabels).

NDA provisions (*NDA*): Finally, we let a lawyer select provisions that typically occur in non-disclosure agreements (NDAs). We select NDA provisions, because we will evaluate the NDA provision classifier on proprietary data stemming from outside our corpus to evaluate the broader applicability of the labeled EDGAR provisions. This yields 46 labels and 110,082 provisions (8.30% with multilabels).[12]

For all data subsets, we perform a random 70%-10%-20% split to obtain training, development, and test set, respectively.

4.2. Classifier selection

For classification, we compare the following approaches:

Label name. Like Chalkidis et al. (2019), we observed that for some provisions, the label name occurs in the provision itself. Thus, we check a provision for all label names and apply all labels whose names we find.

LogReg. We transform the texts into their unigram TFIDF representations and train one logistic regression classifier per label.

BoW+MLP. This approach uses neural embeddings by fasttext (Joulin et al., 2018; Bojanowski et al., 2017) as word representations. We represent provisions as bag-of-words (BoW). The embeddings are then fed into two-layered multilayer perceptron (MLP) for classification (with sigmoid as the last activation function to produce output suited for multilabel classification). To derive fixed-length sentence representations suited for classification, we compare two different aggregation methods:

- **Basic averaging (BoW+MLP)**: We average each dimension of the token embeddings.
- **BoW+MLP+Attn**: Before averaging the token embeddings, we feed them through an attention layer (Yang et al., 2016) which assigns each embedding a weight regarding its usefulness for classification. The weighted sum of the embeddings then serves as the classifier input.

DistilBERT. We fine-tune DistilBERT (Sanh et al., 2019), a distilled version of BERT (Devlin et al., 2018) and use its [CLS] token as the classifier input. We use the reference implementation of DistilBERT and the associated classification fine-tuning script from Huggingface's "Transformers" Library (Wolf et al., 2019) and modify it for multi-label classification (by replacing the softmax with the sigmoid activation).

[9] https://drive.switch.ch/index.php/s/j9S0GRMAbGZKa1A

[10] https://github.zhaw.ch/tuge/LEDGAR_provision_classification

[11] The list of labels is shown in the Appendix.

[12] Cf. the Appendix for the complete list.

For both the word embedding and BERT approaches, we use a two-layer feed-forward neural network as a classifier, with sigmoid activation function at the last layer to obtain multilabel predictions. We fine-tune the classification thresholds for each classifier by evaluating all threshold between 0.1 and 0.9 in 0.01 increments, retaining the one that yields the best micro F1-score for each label.

4.3. Results

Table 4 shows the results of our classification experiments.

proto

	Macro			Micro		
	Rec	Prec	F1	Rec	Prec	F1
Label name	0.62	0.64	0.63	0.60	0.60	0.60
LogReg	0.94	0.91	0.92	0.95	0.92	0.93
BoW+MLP	0.93	0.90	0.92	0.94	0.91	0.93
BoW+MLP+Attn	0.92	0.93	0.93	0.92	0.94	0.93
DistilBERT	0.95	0.94	0.94	0.95	0.95	0.95

freq100

	Macro			Micro		
	Rec	Prec	F1	Rec	Prec	F1
Label name	0.41	0.13	0.20	0.49	0.04	0.07
LogReg	0.75	0.61	0.67	0.78	0.59	0.68
Bow+MLP	0.61	0.72	0.66	0.68	0.70	0.69
BoW+MLP+Attn	0.61	0.73	0.67	0.69	0.72	0.71
DistilBERT	0.63	0.59	0.61	0.67	0.53	0.60

leaves

	Macro			Micro		
	Rec	Prec	F1	Rec	Prec	F1
Label name	0.58	0.11	0.18	0.60	0.07	0.12
LogReg	0.76	0.65	0.70	0.76	0.63	0.69
BoW+MLP	0.61	0.72	0.66	0.67	0.71	0.69
BoW+MLP+Attn	0.64	0.74	0.69	0.70	0.74	0.72
DistilBERT	0.58	0.62	0.60	0.73	0.20	0.31

NDA provisions

	Macro			Micro		
	Rec	Prec	F1	Rec	Prec	F1
Label name	0.14	0.11	0.12	0.23	0.48	0.31
LogReg	0.84	0.84	0.84	0.94	0.90	0.92
BoW+MLP	0.73	0.83	0.78	0.91	0.92	0.92
BoW+MLP+Attn	0.77	0.88	0.82	0.93	0.94	0.93
DistilBERT	0.74	0.82	0.78	0.92	0.92	0.92

Table 4: Results of provision classification.

Overall, we observe that the subcorpora are well-suited for classification, yet the classification task is not easily solved by matching keywords from the labels.

We also observe a distinction between the logistic regression classifier and the neural models: While micro F1-scores are comparable, LogReg features higher recall, but the neural models outperform it in precision. We believe that this difference stems from the different input representations. LogReg has direct access to the tokens through their TFIDF representations, while the neural model all operate on aggregated token representations, i.e. averaged token embeddings, or the `[CLS]` token in the case of DistilBERT. We believe thus that the classification tasks in our subcorpora strongly rely on learning associations between individual keywords and labels. LogReg, with its direct access to keyword occurrences, is able to pick up such associations more quickly with fewer training examples, as evidenced by its strong macro F1 and recall scores. DistilBERT seems to struggle to learn these associations sufficiently for larger labelsets with lower-frequency labels, and our adapted implementation yields poor results on the *leaves* subcorpus. We found that the logits produced by DistilBERT for low-frequency labels are not reliable when tuning the classification thresholds, and the classifier then over-predicts these labels, resulting in a low micro precision. One measure to alleviate this could be to apply an attention layer over the BERT token embeddings to obtain a weighted average for classification, rather than relying on the `[CLS]` token.

Finally, the inclusion of the attentional layer into the BoW-MLP improves performance across the board.

4.4. Application to out-of-domain NDAs

We designed a quasi zero-shot classification or transfer experiment to verify whether LEDGAR is a valid resource to train classifiers for provision classification in contracts that do not stem from SEC filings.

To this end, we used a set of publicly available English NDAs that we gathered from the internet. A group of three legal professionals annotated the NDAs using a proprietary annotation scheme. Within their proprietary labelset, we identified a subset of 46 labels that can be mapped to labels in the LEDGAR provisions based on the label names.[13]

It is noteworthy that the annotators followed specific annotation guidelines by their company, while the labels in the LEDGAR subcorpus were extracted automatically, and were assigned by the creators of the contracts in the SEC filings while drafting the contracts. That is, the labels stem from different sources and were assigned to the provisions in different settings.

Table 5 shows the statistics of the proprietary annotations. One stark difference to the LEDGAR corpus is that the average token count is much lower (30 vs. 124). This difference stems from the units annotated in the corpora: While LEDGAR's labels apply to paragraphs, the proprietary annotations were labeled on the sentence level. Furthermore, in the NDA subcorpus of LEDGAR, 8% of the provisions have mutlilabels, while in the proprietary data, 51% of the provisions have multilabels.

Quantity	amount (std. dev.)
No. of tokens	423,654
Vocabulary size	7,921
No. of provisions	14,142
Multilabel provisions	7,193
Avg. tokens per provision	30 (23)

Table 5: Token and provision statistics of the proprietary annotations.

Again, we apply a 70-10-20% split to obtain train, development, and test set. We compare zero-shot classification to in-domain training, i.e. the setting where the classifiers are trained on the proprietary annotations. For the zero-shot experiments, we tune the classification thresholds on the proprietary development set. Table 4.4. shows the classification results.

Clearly, in-domain training yields better results than zero-shot classification. Using in-domain training, LogReg

[13] Cf. the appendix for the list of NDA-related labels.

NDA proprietary; in-domain training

	Macro			Micro		
	Rec	Prec	F1	Rec	Prec	F1
Label name	0.06	0.09	0.07	0.05	0.16	0.08
LogReg	0.61	0.64	0.62	0.75	0.77	0.76
BoW+MLP	0.50	0.59	0.54	0.69	0.74	0.72
BoW+MLP+Attn	0.51	0.57	0.54	0.66	0.61	0.63
DistilBERT	0.34	0.37	0.35	0.60	0.47	0.53

NDA proprietary; zero-shot

	Macro			Micro		
	Rec	Prec	F1	Rec	Prec	F1
Label name	0.06	0.09	0.07	0.05	0.16	0.08
LogReg	0.47	0.46	0.46	0.57	0.42	0.48
BoW+MLP	0.33	0.43	0.37	0.38	0.55	0.45
BoW+MLP+Attn	0.37	0.47	0.41	0.44	0.51	0.47
DistilBERT	0.44	0.44	0.44	0.55	0.48	0.52

Table 6: Classification results on proprietary NDA annotations.

achieves the best results. Also, the attention layer decreases performance for BoW+MLP. We assume that the proprietary annotations do not contain enough data for several labels in order to train the attention weights properly. DistilBERT also fails, as in the *leaves* subcorpus, and we again found that the low-frequency labels are the culprit. In the zero-shot results, however, where enough samples are present in the LEDGAR corpus to train the model, DistilBERT achieves the highest micro F1 scores, while LogReg outperforms the other classifiers in terms of macro F1 based on its strong recall.

We inspected the evaluation reports of the individual provision types in the zero-shot setting for DistilBERT to identify whether all provision types perform worse, or if the drop in micro/macro F1 score mainly stems from the lower performance on specific labels. Indeed, we found that the F1 scores vary strongly for the different labels. Firstly, we found a moderate to strong correlation between frequency of the labels and their F1 score (pearson's r of 0.44). Secondly, for several provision types, performance is high (around 0.80 F1 score for e.g. *Amendments, Battle of Forms, Counterpart Copy, Jurisdiction, Warranty, Non-Assignment, Severability, Waivers*), while for others, especially low-frequency labels, transferring the fine-tuned DistilBERT fails. Upon inspecting samples from the two corpora, we found that the semantics of some labels differ in the two corpora, although the label name suggests a high similarity (F1 scores of around 0.30 for e.g. *Arbitration Choice of Law, Consequences of Termination of Contract, Data Security, Liquidated Damage, Warranty of Title*). For example, the consequences of terminating a contract vary across contract types, such as NDAs (one might have to destroy confidential data), software license agreements (licensed software needs to be uninstalled), or construction agreements (certain costs might have to be redeemed). Our classifiers were trained on provisions stemming from different contract types in the LEDGAR corpus, but in the proprietary annotations, these provisions have a more restricted, NDA-specific meaning.

Based on these results, we believe that our corpus is useful as a resource when developing classifiers in the legal domain for contract analysis. It offers various provision types that, in our experiments, are suitable for direct transfer to contracts stemming from outside the corpus. However, we suggest verifying whether the definitions of the provisions align with the intended use. Some provision types (e.g. *Consequences of Termination of Contract*) do seem to inherit specific semantics in the different contract types they occur in. Selecting only provisions from contract types of the target domain may increase classifier performance for such provision types.

5. Conclusion

We have presented a freely-available corpus for text classification in the domains of Legal NLP and large-scale multilabel, multiclass classification. We have demonstrated different subsampling approaches to obtain subcorpora and performed classification experiments on them. The experiments showed that the corpus is well-suited for both research in Legal NLP, as well as large-scale multilabel classification.

Future work will include extending the label merging steps, as we found several labels with seemingly equivalent semantics (e.g. *withholding taxes, withholding of tax, tax withholding*), as well as applying classification approaches from the extreme multiclass classification domain (Jain et al., 2019, e.g.) on the full corpus. Finally, extracting contract types and assigning extracted provisions to their contract types might increase the specificity of the extracted provisions when targeting certain types of contracts for provision classification.

6. Acknowledgements

Work on this paper was funded by Innosuisse (formerly Kommission für Technologie und Innovation KTI) under grant number 28148.1 PFES-ES.

Appendix

Prototypical provision labels

Labels selected for the *proto* subcorpus:
Amendments, Assignments, Assigns, Counterparts, Entire Agreements, Expenses, Governing Laws, Notices, Severability, Successors, Survival, Terminations, Waivers

NDA provision labels

Labels selected for the *NDA* subcorpus:
Amendments, Arbitration Choice of Law, Arbitration Procedure, Audit, Battle of Forms, Change of Control, Confidential Information, Consequences of Termination of Contract, Counterpart Copy, Data Privacy, Data Security, Definitions, Duration of Contract, Duty of Confidentiality Mutual, Equitable Remedies, Expenses, Export Restrictions, Extra Costs, Governing Law, IP Rights, Jurisdiction (Courts), Legal Costs, Liability Exclusion, Liability Reasons, License Rights, Liquidated Damages, Mediation, No Third Party Rights or Beneficiaries Disclaimer, No Warranty, Non-Assignment, Non-Solicitation Agreement, Notice Date, Notices, Ordinary Notice Period, Ownership, Passing of Risk, Payment Deadline, Permissible Disclosure to Third Party, Place of Performance, Reference to Further Agreements, Renewal or Extension of Contract, Severability, Succession, Termination of Contract, Waivers, Warranty of Title

7. Bibliographical References

Nikolaos Aletras, et al., editors. (2019). *Proceedings of the Natural Legal Language Processing Workshop 2019*, Minneapolis, Minnesota. Association for Computational Linguistics.

Bengio, S., Dembczynski, K., Joachims, T., Kloft, M., and Varma, M. (2019). Extreme Classification (Dagstuhl Seminar 18291). *Dagstuhl Reports*, 8(7):62–80.

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

Bommarito, M. J., Katz, D. M., and Detterman, E. M. (2018a). LexNLP: Natural language processing and information extraction for legal and regulatory texts. *arXiv preprint*, abs/1806.03688.

Bommarito, M. J., Katz, D. M., and Detterman, E. M. (2018b). OpenEDGAR: Open source software for SEC EDGAR analysis. *arXiv preprint*, abs/1806.04973.

Chalkidis, I., Fergadiotis, E., Malakasiotis, P., and Androutsopoulos, I. (2019). Large-scale multi-label text classification on EU legislation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6314–6322, Florence, Italy, July. Association for Computational Linguistics.

Choromanska, A. and Kumar Jain, I. (2019). Extreme multiclass classification criteria. *Computation*, 7(1).

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

Fisch, J. E. (2018). Governance by contract: The implications for corporate bylaws. *Calif. L. Rev.*, 106:373.

Glaser, I., Scepankova, E., and Matthes, F. (2018). Classifying semantic types of legal sentences: Portability of machine learning models. In *JURIX*, pages 61–70.

Hershkoff, H. and Kahan, M. (2018). Selection provisions in corporate contracts. *Wash. L. Rev.*, 93:265.

Jain, H., Balasubramanian, V., Chunduri, B., and Varma, M. (2019). Slice: Scalable linear extreme classifiers trained on 100 million labels for related searches. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 528–536. ACM.

Joulin, A., Bojanowski, P., Mikolov, T., Jégou, H., and Grave, E. (2018). Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Jeroen Keppens et al., editors. (2017). *Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law, ICAIL 2017*. ACM.

Knapp, C. L., Crystal, N. M., and Prince, H. G. (2019). *Problems in Contract Law: cases and materials*. Aspen Publishers.

Mills, K. R. (2019). Talking about regulation in 10-k annual reports; uniformity in a naive sample. *Finance Graduate Theses & Dissertations*.

Monica Palmirani, editor. (2018). *Legal Knowledge and Information Systems - JURIX 2018: The Thirty-first Annual Conference, Groningen, The Netherlands, 12-14 December 2018*, volume 313 of *Frontiers in Artificial Intelligence and Applications*. IOS Press.

Georg Rehm, et al., editors. (2018). *Proceedings of the 1st Workshop on Language Resources and Technologies for the Legal Knowledge Graph*, Paris, France. European Language Resources Association (ELRA).

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Soni, A., Pappu, A., Busa-Fekete, R., and Dembczynski, K. (2018). Extreme multilabel classification for social media chairs' welcome and organization. In *Companion Proceedings of the The Web Conference 2018*, pages 1893–1894. International World Wide Web Conferences Steering Committee.

Trosborg, A. (1991). An analysis of legal speech acts in english contract law. 'It is hereby performed'. *HERMES-Journal of Language and Communication in Business*, 4(6):65–90.

Trosborg, A. (1995). Statutes and contracts: An analysis of legal speech acts in the english language of the law. *Journal of Pragmatics*, 23(1):31–53.

Waltl, B., Muhr, J., Glaser, I., Bonczek, G., Scepankova, E., and Matthes, F. (2017). Classifying legal norms with active machine learning. *Legal Knowledge and Information Systems*, page 11.

Waltl, B., Bonczek, G., Scepankova, E., and Matthes, F. (2019). Semantic types of legal norms in german laws: classification and analysis using local linear explanations. *Artificial Intelligence and Law*, 27(1):43–71.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). Transformers: State-of-the-art natural language processing.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., and Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Yovel, J. (2000). What is contract law about–speech act theory and a critique of skeletal promises. *Northwestern University Law Review*, 94:937.