Long-length Legal Document Classification

Abstract

One of the principal tasks of machine learning with major applications is text classification. This paper focuses on the legal domain and, in particular, on the classification of lengthy legal documents. The main challenge that this study addresses is the limitation that current models impose on the length of the input text. In addition, the present paper shows that dividing the text into segments and later combining the resulting embeddings with a BiLSTM architecture to form a single document embedding can improve results. These advancements are achieved by utilising a simpler structure, rather than an increasingly complex one, which is often the case in NLP research. The dataset used in this paper is obtained from an online public database containing lengthy legal documents with highly domain-specific vocabulary and thus, the comparison of our results to the ones produced by models implemented on the commonly used datasets would be unjustified. This work provides the foundation for future work in document classification in the legal field.

Introduction

Text classification is a problem in library, information and computer science and one of the most classical and prominent tasks in Natural Language Processing (NLP). In particular, document classification is a procedure of assigning one or more labels to a document from a predetermined set of labels. Automatic document classification tasks can be divided into three categories: supervised, unsupervised and semi-supervised. This study focuses on supervised document classification.

Research so far has focused on short text BIBREF0, BIBREF1, BIBREF2, BIBREF3, whereas the main objective of this paper is to address the classification of lengthy legal documents. In fact, pre-existing

models could not be applied on our corpus, which consists of excessively lengthy legal documents. In the legal field, companies manage millions of documents per year, depending on the size of the company. Therefore, automatic categorisation of documents into different groups significantly enhances the efficiency of document management and decreases the time spent by legal experts analysing documents.

Recently, several quite sophisticated frameworks have been proposed to address the document classification task. However, as proven by BIBREF3 regarding the document classification task, complex neural networks such as Bidirectional Encoder Representations from Transformers (BERT; BIBREF4) can be distilled and yet achieve similar performance scores. In addition, BIBREF5 shows that complex architectures are more sensitive to hyperparameter fluctuations and are susceptible to domains that consist of data with dissimilar characteristics. In this study, rather than employing an overly complex neural architecture, we focus on a relatively simpler neural structure that, in short, creates text embeddings using Doc2Vec BIBREF6 and then passes them through a Bi-directional LSTM (BiLSTM) with attention before making the final prediction.

Furthermore, an important contribution of this paper to automatic document classification is the concept of dividing documents into chunks before processing. It is demonstrated that the segmentation of lengthy documents into smaller chunks of text allows the context of each document to be encapsulated in an improved way, leading to enhanced results. The intuition behind this idea was formed by investigating automatic audio segmentation research. Audio segmentation (also known as audio classification) is an essential pre-processing step in audio analysis that separates different types of sound (e.g. speech, music, silence etc.) and splits audio signals into chunks in order to further improve the comprehension of these signals BIBREF7. Analogously, the present paper shows that splitting overly lengthy legal documents into smaller parts before processing them, boosts the final results.

Related Work

In several industries that produce or handle colossal amounts of text data such as the legal industry, document categorisation is still often performed manually by human experts. Automatic categorisation of documents is highly beneficial for reducing the human effort spent on time-consuming operations. In particular, deep neural networks have achieved state-of-the-art results in document classification over the last few years, outperforming the human classifiers in numerous cases.

Related Work ::: Document Classification Datasets

The majority of researchers evaluate their document classifying models on the following four datasets: Reuters-21578 BIBREF8, ArXiv Academic Paper Dataset - AAPD BIBREF9, IMDB reviews BIBREF10, and Yelp 2014 reviews BIBREF11. However, these commonly used datasets do not contain large documents, which conflicts with one of the main objectives of this study. Note that our definition of `document' in this specific context is a document that has at least 5000 words.

For that purpose, we use a dataset provided by the U.S Securities and Exchange Commission (SEC), namely EDGAR BIBREF12. As anticipated, most models that have achieved inspiring results have very poor performance or even fail when they are tested on large documents from the EDGAR corpus. As shown in Table TABREF1 and Table TABREF3, the differences between the commonly used datasets and the EDGAR dataset are evident.

Related Work ::: Document Classification Approaches

The application of deep neural networks in the field of computer vision has achieved great success. Following this success, several well-known DNN models attained remarkable results when applied on the document classification task. One of the most popular models is the Hierarchical Attention Network (HAN) proposed by BIBREF0. HAN used word and sentence-level attention in order to extract meaningful

features of the documents and ultimately classify them. However, the fact that this architecture is based on a Gated Recurrent Unit (GRU) framework combined with the excessive size of the documents in our corpus would severely affect the results. Concretely, using overly large documents would result in a vast number of time steps and the vanishing gradient problem would be detrimental to performance.

A different yet powerful framework, namely BERT BIBREF4, has achieved state-of-the art results on a large amount of NLP tasks. BERT architecture employs self-attention instead of general attention, thus making the neural network even more complex. Nevertheless, BIBREF3 have established groundbreaking results and demonstrated that sophisticated architectures such as BERT are not necessary to succeed in the document classification task. Furthermore, it is worth mentioning that both the aforementioned models were trained on a rather different corpora. The main difference between the datasets used by those researchers and the EDGAR dataset is the size of the documents, which explains why these models could not be utilised in the present study. In particular, BERT was incompatible with our dataset due to the maximum input sequence length that imposes, namely the 512 terms threshold.

Methods

The novelty of this work is the application of audio segmentation used for speech recognition BIBREF13 in document classification. The ultimate purpose of audio segmentation is to divide the signal into segments, each of which contains distinct audio information. In our case, the same occurs during the document segmentation, where the split chunks become the inputs of our neural network.

From a human perspective, when reading a rather long document or book, we are constantly storing and updating our memory with the essential parts or information of that record. Once enough information is stored in our memory we can form connections so as to gain a deeper understanding of the context and potentially extract valuable insight. In the same way, instead of passing the whole document to Doc2Vec,

we split the document into multiple chunks (Figure FIGREF5). Hence, the machine can imitate human behaviour by identifying and determining the relevance of each chunk.

We create different models with respect to the number of chunks that we divide the initial text into, in order to observe how the different number of chunks affect the efficiency of the final model. These chunks are then used to train Doc2Vec. In short, the intuition behind Doc2Vec is analogous to the intuition behind Word2Vec, where the

words are used to make predictions about the target word (central word). The additional part of Doc2Vec is that it also considers the document ID when predicting a word. Ultimately, after the training each chunk has the form of an embedding.

In the next phase, we aggregate the different chunk embeddings of a document into one vector through the use of a BiLSTM (see Figure FIGREF10). First, the different chunk embeddings $E_{i}^1, E_{i}^2,..., E_{i}^n$ of a document are sequentially fed to the BiLSTM model. Then, the outputs of the forward and the backward layer are concatenated; $h_{it}=[\overrightarrow{h_{it}}\overleftarrow{h_{it}}]$. h_{it} denotes the resulting vectors.

The final classification is subjected to the various features that each chunk contains. Thus, the attention mechanisms are introduced so as to enable the assignment of different weights to each chunk, depending on how strong of a class indicator this chunk is. In particular, the attention scores are assigned to the corresponding hidden state outputs as follows:

Here $\alpha _{it}$ is the attention score assigned to hidden state h_{it} of document i at time step t. This score is determined by the similarity between u_{it} and u_{w}, where u_{it} is a mere non-linear transformation of h_{it} and u_{w} is the context (category) vector BIBREF1. During the

following steps, the products of the hidden states and their corresponding attention scores are calculated and the document vector d_{i} is formed from the summation of those products. Note that u_{w} is randomly initialised and then constantly updated during the training process.

Ultimately, we try different classifiers in order to assess the impact of the segmentation method. As part of the models of the first type, the resulting document vector is output from a batch normalisation layer. A linear transformation is then applied to that and this output is passed through a softmax classifier in order to acquire the multi-class probabilities. This final process is summarised in the following formula:

where $W\in R^{c\times d}$ is the weight matrix, c and d are the number of classes and the number of dimensions of the hidden states respectively and $b\in R^d$ is the bias term. Hence, the final vector s_{i} is a c-dimension vector comprising the probability of that document belonging to each class.

The models of the second type are based on a strong machine learning classifier, namely Support Vector Machine (SVM). SVM also performs document classification by utilising the resulting document embeddings. The main parameters used to train SVM were obtained by optimising each model separately (see Section SECREF14).

Experimental Setup

We evaluate the proposed model on a document classification dataset; 70% of the data is used for the training and the remaining 30% is equally divided and used for tuning and testing our model.

During the pre-processing stage where the documents are split into chunks, we utilise a cluster of Azure Virtual Machines with 32 GB RAM and 16 cores, which are optimised for CPU usage. A similar cluster is used during the hyperparameter optimisation, however, with 112 GB RAM. Reading from the remote disk

(Azure Blob Storage) is rather time-consuming, since the corpus comprises lengthy documents. Thus, to accelerate the training, we chose nodes with abundant memory in order to load everything in memory just once (required roughly one hour for that process).

We use Pytorch 1.2.0 as the backend framework, Scikit-learn 0.20.3 for SVM and dataset splits, and gensim 3.8.1 for Doc2Vec model.

Regulation S-K is an official regulation under the US Securities Act of 1933 that establishes reporting regulations for a variety SEC filings used by public companies.

Experimental Setup ::: Dataset

The data we use to evaluate our model is a set of documents downloaded from EDGAR, an online public database from the U.S. Securities and Exchange Commission (SEC). EDGAR is the primary system for submissions by companies and others who are required by law to file information with the SEC. These documents can be grouped according to filing types, which determines the substantial content to fulfill their filing obligation. To work on as many documents as possible, we choose the following types: "10-Q", "10-K", "EX-99.1", "EX-10.1" and "EX-101.INS". The total number of documents is 28,445 and there are 5,689 documents for each filing type. We summarise the statistics of this dataset in Table TABREF11.

Almost all documents of type "10-K" begin with lines that contain identical headings. In order to enable the machine to truly comprehend why a document of type "10-K" should be categorised to that filing type, we remove the first six lines where the identical text is located. The model is then able to focus on finding common features that exist in documents of the same filing type, rather than focusing on just capturing the few sentences that are the same in almost all of the documents of type "10-K". A similar procedure is followed with the documents of type "10-Q".

Experimental Setup ::: Model Configuration

As Table TABREF13 shows, we create seven different models that correspond to the number of chunks that the text is divided into before passing through Doc2Vec. Each model is optimised separately to ensure fair comparison.

For the optimisation of the BiLSTM with attention model, we use Adam optimiser with a learning rate of 0.001, batch size of 1,000 and distinct values for each one of the other hyperparameters. Analogously, the SVM classifier consists of the Radial Basis Function (RBF) as the kernel function and a different value of gamma and the penalty parameter for each different model. The intention of the distinct values used for each model is to optimise each model separately so as to enable them to reach their best performance.

Furthermore, we observe that Doc2Vec requires only a small portion of the corpus to train accurately. Indeed, when training Doc2Vec on more documents we observe a substantial decrease in accuracy. It is well-known that legal documents contain several domain-specific words that are often repeated not only among different documents, but also within the same document. Training Doc2Vec on more documents introduced undesirable noise that results from company names, numbers such as transaction amounts and dates, job titles and addresses. Consequently, Doc2Vec is proven to generate more accurate document embeddings when trained on just 150 randomly chosen documents (30 for each filing type).

Results and Discussion

Recently, reproducibility is becoming a growing concern for the NLP community BIBREF14. In fact, the majority of the papers we consider in this study fail to report the validation set results. To address these issues, apart from the F1 scores on the test sets we also report the F1 scores for the validation sets.

Legal documents contain domain-specific vocabulary and each type of document is normally defined in a very unambiguous way. Hence, even simple classifiers can achieve relatively high accuracy when classifying different documents. Nevertheless, even the slightest improvement of 1% or less will result in the correct classification of thousands of additional documents, which is crucial in the legal industry when handling large numbers of documents. This research allows these simple classifiers to achieve even greater results, by combining them with different architectures.

As Table TABREF13 and Table TABREF15 indicate, dividing the document in chunks - up to certain thresholds - results in improved models compared to those where the whole document is input into the classifier. Note that the model with one chunk denotes the model which takes as input the whole document to produce the document embedding and thereby is used as a benchmark in order to be able to identify the effectiveness of the document segmentation method.

More specifically, splitting the document into chunks yields higher test accuracy than having the whole document as input. Our first model with the BiLSTM based framework and the linear classifier reaches a 97.97% accuracy with a 1.1% improvement upon the benchmark model. Similarly, the second model with the SVM classifier reaches a remarkable 98.11% accuracy with a 0.4% improvement upon the benchmark model.

A more thorough investigation of the test accuracy scores indicate that documents of type "EX-99.1" are the ones that get misclassified the most, whereas the remaining four types of documents are in general classified correctly at a considerably higher rate. As confusion matrix plot in Figure FIGREF18 highlights, there are cases that documents of type "EX-10.1" are misclassified as "EX-99.1", however, the reverse occurs more frequently. Further exploration of documents of type "EX-99.1" reveals that these documents often contain homogeneous agreements or clauses with the ones embodied in documents of type "EX-10.1".

Ultimately, Figure FIGREF16 and Figure FIGREF17 demonstrate the increase of the efficiency of the document embeddings after the use of BiLSTM. These vector representations of each cluster have noticeably more robustly defined boundaries after they are passed through the BiLSTM network compared to the ones that are only passed through the mere Doc2Vec.

Conclusion

The main contribution of this paper is to overcome the document length limitations that are imposed by most modern architectures. It also shows that dividing documents into chunks before inputting them into Doc2Vec can result in enhanced models. Nonetheless, these advancements are accomplished with a relatively simplified structure, rather than a significantly more sophisticated architecture than its predecessors, which is often the case in NLP research.

One potential extension of this work would be to apply powerful yet computationally expensive pre-processing techniques to the various documents. Techniques such as Named Entity Recognition (NER) could enable the training of the whole corpus in Doc2Vec by removing the undesired noise. Furthermore, the projections of the document embeddings at the end of our pipeline are shown to have clearly defined boundaries and thus they can be valuable for different NLP tasks, such as estimating document similarities. In the legal industry, this can contribute to identifying usages of legal templates and clauses.