Hovedoppgave

NTNU
Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

Henrik Hodnefjeld

Classifying European Court of Human Rights cases using transformer based models

Hovedoppgave i Institutt for datateknologi og informatikk
Veileder: Ali Shariq Imran
Medveileder: Zenun Kastrati
Juni 2022

NTNU
Kunnskap for en bedre verden

Henrik Hodnefjeld

Classifying European Court of Human Rights cases using transformer based models

Hovedoppgave i Institutt for datateknologi og informatikk
Veileder: Ali Shariq Imran
Medveileder: Zenun Kastrati
Juni 2022

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for datateknologi og informatikk

NTNU

Kunnskap for en bedre verden

Abstract

Models based on transformers, such as Bidirectional Encoder Representation from Transformers (BERT), have traditionally not been applied to text sequences with greater length than that of sentences. Such models are computationally expensive and require a large amount of memory. Transformer-based models are also often pre-trained on generalized languages, which makes them less effective in language-specific domains, such as legal documents.

In the field of natural language processing, there is a growing interest in creating newer models that can handle more complex input sequences and domain-specific languages. This work builds upon the previous efforts made within the domain of document classification, specifically concerning legal contexts.

Based on our examinations of different models we propose using a sliding window approach to increase the normal maximum sequence length of models. Our results are validated by thorough empirical experiments in which we outperform previous results on similar tasks.

Sammendrag

Modeller basert på transformers, som Bidirectional Encoder Representation from Transformers (BERT), har tradisjonelt ikke blitt brukt på tekstsekvenser med større lengde enn setninger. Å kjøre slike modeller har høye driftskostnader og krever mye minne. Slike transformer baserte modeller er ofte forhåndstrent på generaliserte språk, noe som gjør dem mindre effektive i språkspesifikke domener, for eksempel juridiske dokumenter.

Innenfor fagfeltet for naturlig språkbehandling er det en økende interesse for å lage nyere modeller som kan håndtere mer komplekse inputsekvenser og domenespesifikke språk. Arbeidet i denne masteroppgaven bygger på den tidligere innsatsen som er gjort innen dokumentklassifisering, med fokus på det juridiske språket.

Basert på våre undersøkelser av forskjellige BERT-modeller foreslår vi å bruke et bevegende skyvevindu tilnærming for å øke den normale maksimale sekvenslengden på modellene. Resultatene våre er validert av grundige empiriske eksperimenter der vi utkonkurrerer tidligere resultater på lignende oppgaver.

Acknowledgements

This thesis finalizes my master's degree in computer science at the Norwegian University of Science and Technology, specializing in artificial intelligence. I want to preface this with some acknowledgements.

First and foremost, I'd like to express my gratitude to Professor Ali Shariq Imran of the Department of Computer Science at NTNU and Zenun Kastrati at Linnaeus University, who served as my thesis advisors. Throughout my thesis, Ali and Zenun have been invaluable resources. Their assistance enabled me to engage in an intriguing research topic by pointing me in the appropriate direction while helping my thesis to be the result of my own efforts and motivation.

Secondly, I want to thank the team at the legal Tech-startup Etain for being my guide in the legal world. Finally I want to extend my gratitude to my friends and family for their continuous support and encouragement during my academic career. Thank you.

Contents

Tables

Figures

Chapter 1

Introduction

In this chapter, we will provide a brief overview of the problem that this thesis intends to investigate and the rationale for its research. Additionally, this chapter covers the research questions we aim to answer as well as the contributions this thesis makes.

1.1 Background and Motivation

We live in a society increasingly governed by legal rules and regulations (Stortinget 2015). The 'juridification' of society increases the importance of defending and/or upholding one's legal rights. High-quality legal representation is often expensive, so those who cannot afford it rely on public legal aid programs (Carlin and Howard 1964).

Studies of several countries' legal systems indicate that legal aid is not being provided to all those in need. An analysis of the legal system in Norway (Tønnessen 2020) shows that only approximately 9 % of the population is qualified for legal aid.

A study of the legal system in the U.K. (Hirsch 2018) indicates that there are several issues with providing free legal aid . At the maximum level of disposable income at which legal aid is available, many households do not have sufficient income to meet a minimum standard of living before legal fees are paid. Typically, their disposable incomes are 10 to 30 percent too low to meet a minimum budget.(Hirsch 2020)

Even those with the lowest incomes, the most vulnerable individuals are excluded from legal aid if they have savings or assets worth more than £8,000, or in some cases £3,000. If a person has this much money in the bank, he or she can pay for some legal expenses without affecting their current ability to maintain a minimum standard of living (Hirsch 2018).

A means test is used to determine whether a person or household is eligible for a particular benefit or payment (Chaturvedi and Koul 2019). However, the means test also considers the value of people's homes. As a result, homeowners who are not employed may be excluded from legal aid, despite the fact that they have no realistic option for paying the legal fees (Hirsch 2018).

However, by improving the effectiveness of the provision of legal assistance, the cost of legal aid may be reduced, and increase the number of cases assisted within the publicly funded legal aid budgets. Through this thesis, we aim to look at a solution to improve the efficiency, and thereby lowering the costs for private plaintiffs and defendants, of the legal system by automating the classification of legal cases.

We intend to use several transformer-based models to assist with the classification of large legal documents regarding human rights violations. With transformer-based architectures, it has become easier to build more capable models, and pre-training has made it possible to use this capability effectively for a wide variety of tasks.

1.2 Goals and Research Questions

The purpose of this thesis is to investigate the possibility of classifying large legal documents using transformer-based models to determine whether a human rights article has been violated and, if so, which articles. . As a way of determining the optimal approach for categorizing documents of this type, both binary and multilabel methods will be investigated. More specifically, we intend to examine the following research questions (RQs):

RQ1

How effective are transformers in handling long sequences of text data from legal documents?

RQ2

Which features can be exploited to train transformers and effectively classify legal documents?

RQ3

How viable would such a solution be in enhancing the efficiency of legal assistance?

1.3 Contributions

We can summarize our contributions into three points:

- We demonstrate first that it is possible to increase the maximum sequence length for transformer models artificially. With our approach, we use a sliding window technique to allow for multiple sub-sequences to enable the models to evaluate text sequences longer than their usual limit.
- As a second step, we examine potential additions to the text sequence to increase the effectiveness of the models.
- Lastly, we evaluate the performance of both neural and non-neural models for long sequence text classification.

1.4 Thesis Structure

Section 1: Introduction

This section contains the introduction to the problem this thesis aims to solve and its motivation. The research questions and contributions are also covered in this section.

Section 2: Background

This section covers required knowledge to understand the models and language in the rest of the thesis.

Section 3: Related Work

The preliminaries of the thesis are discussed in this section, as well as the relevant methods.

Section 4: Methods and Implementation

This section contains an explanation of the models dataset and the system for scraping more data.

Section 5: Experiments and Results

This section describes the experimental process used and which experiments that were conducted. Furthermore, this section contains the results from these experiments.

Section 6: Discussion

This section will discuss the results from the experiments and results section. We perform a deeper analysis of the results as well as looking at potential limitations to our approach.

Section 7: Conclusion and Further Work

Concludes all of the work done and suggests further work that could follow this work.

Chapter 2

Background

This chapter covers the background theory needed to explain the following methods and implementations in this paper. Firstly looking at how to evaluate models. Then going over the deep learning models relevant for this paper. Thereafter explaining appropriate conventional machine learning techniques. Finally, looking at prior research done within the field of legal document classification.

2.1 Evaluation of Models

In this section we will go over different ways to evaluate how well a model performs a task. This includes using: Accuracy, precision, recall and F-measure.

2.1.1 Specificity and Sensitivity

There are two kinds of positives and two kinds of negatives when it comes to evaluating classification models:

1. True positive (TP) - The model correctly predicts a positive class.
2. False positive (FP) - The model incorrectly predicts a positive class
3. True negative (TN) - The model correctly predicts a negative class.
4. False negative (FN) - The model incorrectly predicts a negative class

A confusion matrix, as seen below, is a technique for summarizing the performance of a classification algorithm using TP, TN, FP, and FN.

		Prediction outcome p	**Prediction outcome** n	**Prediction outcome** total
actual value	**p′**	True Positive	False Negative	P′
	n′	False Positive	True Negative	N′
	total	P	N	

Table 2.1: Confusion Matrix

2.1.2 Accuracy

Accuracy shows how the model is at predicting a correct category (Yin et al. 2019). Accuracy works well in scenarios with a balanced dataset. However, if 90% of our dataset can be categorized as a true positive, a 90% accuracy would result from the model blindly predicting true positives.

$$Accuracy = \frac{TruePositives + TrueNegatives}{AllPredictions(TP + TN + FP + FN)} \tag{2.1}$$

Accuracy is calculated by the formula seen in figure2.1. The formula takes all the model's correct predictions and divides this by the total number of samples.

2.1.3 Precision and Recall

The precision score gets the number of relevant predictions from the model's total number of positive predictions.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{2.2}$$

Formula 2.2 shows how precision score is calculated by dividing the number of true positives by the number of positive predictions. (Buckland and Gey 1994).

With this, one can tell how many of the predictions are relevant. A 100% precision score would mean no false positive predictions, and every positive prediction from the model was correct.

Recall shows the number of cases the model correctly classified as relevant out of all relevant cases.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives} \tag{2.3}$$

The recall score is calculated by true positives divided by the number of predictions that should have been positive (see figure 2.3) . A 100% recall score would mean no false negatives predicted. Therefore, all negative predictions are correct.

2.1.4 F1-score

F1-score is the cumulative result of all previous sections. Combining the precision and recall score into a single metric (Davis and Goadrich 2006). F1-score is primarily used for comparison between classifiers. This is because having a good recall score does not guarantee a good precision score and vice versa.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.4}$$

F1-score is calculated by the formula seen in equation 2.4. F1-score results in a harmonic mean between the two measuring units (Opitz and Burst 2019).

2.2 Data

Machine learning model's end goal is to make correct predictions on new unknown data not seen in training. Therefore we want our algorithms to train on huge amounts of data to be able to learn relations between existing data to easily identify relations with new data. However, this can lead to the resulting models having learnt biases from the data it is trained on that does not correlate with new data. In this section, we will look at some of the pitfalls and solutions for this problem.

2.2.1 Overfitting and Underfitting

Having a generalized model exempt from biases requires your model to not be overly biased towards the training data you have used while at the same time having learnt enough from the training data to catch patterns. These two scenarios are called overfitting and underfitting.

The reason for overfitting is often due to a lack of data (Ying 2019). As the sample size of data decreases, so does the potential patterns the algorithm can pick up on. This might lead to the final model having picked up patterns that don't exist at all outside of the training dataset.

The following are some of the strategies that has proposed to reduce the effects of overfitting:

1. Early-stopping is introduced to prevent overfitting by stopping training before the performance stops optimizing. (Raskutti et al. 2014).
2. A "network-reduction" strategy eliminates noise in the training set.(P. Clark and Niblett 1989)
3. A "data expansion" strategy is proposed for complicated models to fine-tune the hyperparameters sets with sufficient data (Yip and Gerstein 2009).
4. A "regularization" strategy is proposed to guarantee model performance to a large extent while addressing real-world issues by selecting and separating more practical and less practical features (Warde-Farley et al. 2013).

Underfitting happens when the algorithm used to build the prediction model is very simple and not able to learn the complex pattern from the training data (Jabbar and Khan 2015). In that case, accuracy will not only be low when introduced to new data, but also when testing on data from the original dataset.

2.2.2 Scraping

Scraping is the process of extracting data from a website. This can be done manually but is most commonly done automatically. The data can then be stored in a table or CSV for ease of access.

The process of scraping loads in the URL of a web page before accessing that page's HTML, CSS and JS elements to read off data from the page. This can be the whole page or certain parts of the elements and children from the page.

2.2.3 Data augementation

To solve the overfitting issue in general, one needs more data. But if there is no more data available to scrape or easily acquire, there is a third option. Data augmentation. Data augmentation are techniques used to generate additional manufactured data from the data you already have. Data augmentation is often applied in natural language processing and computer vision cases. However, you need a good number of unique data samples to avoid re-generating the same data over and over.

2.3 Conventional Machine Learning techniques

According to Arthur Samuel, machine learning is defined as "computers ability to learn without being explicitly programmed". Conventional machine learning algorithms are based on learning from a training set to develop a trained model for further prediction (Chalkidis, Fergadiotis, Malakasiotis et al. 2020). This section goes over popular conventional machine learning techniques for natural language processing.

2.3.1 SVM

Support Vector Machine (SVM) is a supervised machine learning algorithm. SVMs are focused on finding the hyperplane that divides a dataset into classes seen in figure 2.1. One finds support vectors to find the hyperplane or "line" that separates the dataset.

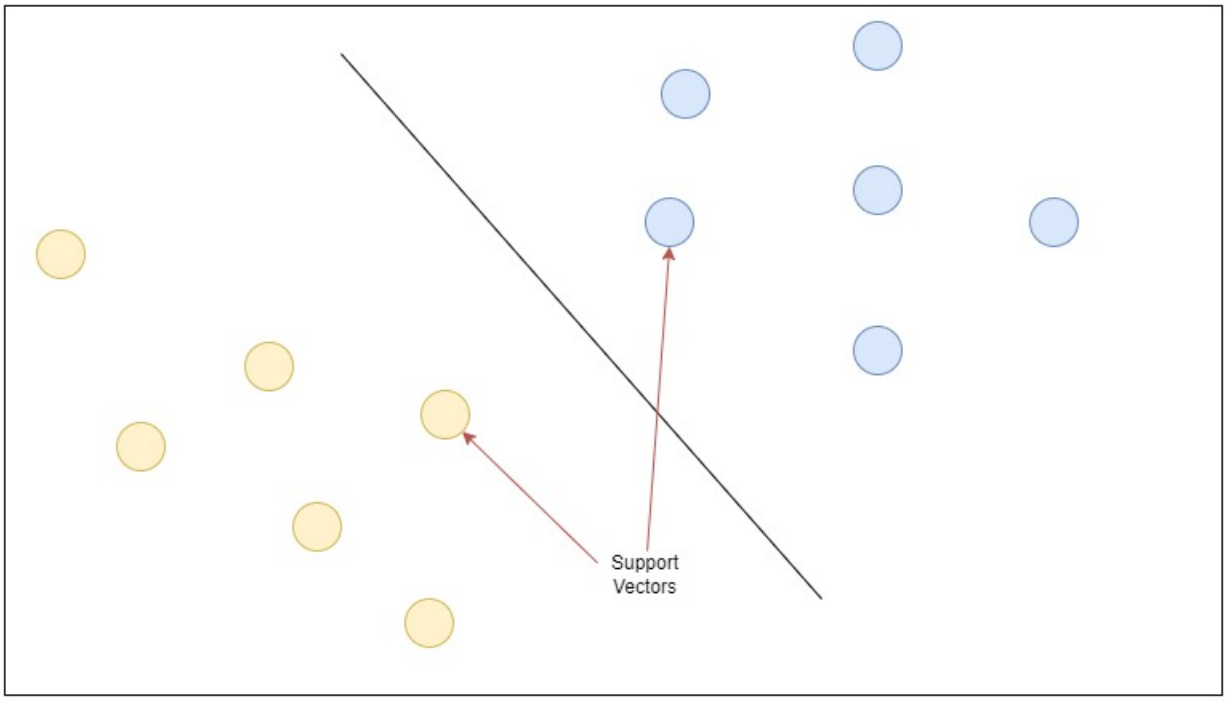


Figure 2.1: Support vector model

Support vectors which give SVM its namesake are the two data points closest to the hyperplane. A new hyperplane would have to be drawn if one of the two support vectors were removed from the dataset. After the hyperplane has been found, one can confidently say that the further away a dataset is from the hyperplane, the more confident we are in its class.

2.3.2 Decision Tree classifier

Decision tree is a supervised machine learning algorithm that divides datasets based on rules that closely resemble human decision-making. We will focus on the classification part of decision trees; however, Decision trees can be used for regression tasks as well. To classify the dataset, a decision tree asks many yes/no questions to split up the dataset continuously. Doing this split results in a tree structure seen in figure 2.2. Every question branches the tree deeper and creates new nodes. When the algorithm stopes asking questions your left with the leaf nodes.

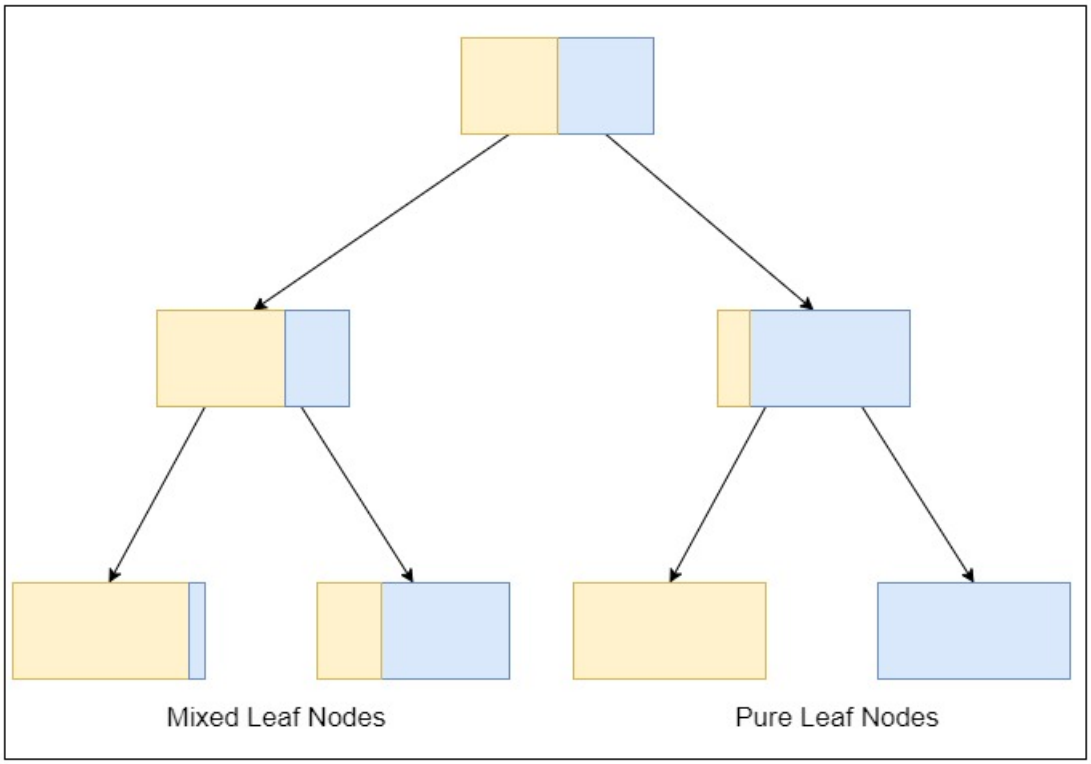


Figure 2.2: Decision tree classifier

Repeating this process until the algorithm does not have any more rules to apply. Afterwards, one can assign classes to the leaf nodes to evaluate what data belongs in what class. The end results can be one of two leaf nodes: Pure leaf nodes and mixed leaf nodes. Pure leaf nodes are the leaf nodes that only belong to one class, and all data that ends in this node can be assigned the leaf nodes class. However, this is often not the case as most leaf nodes are a mix of the classes. In the case of the mixed leaf nodes the majority class is chosen.

2.3.3 Naive Bayes classifier

Naive Bayes bases itself on the Bayes' Theorem: to find the probability of an event occurring given the likelihood of another event that has already happened. The Bayes' theorem is shown in the equation below 2.5.

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{2.5}$$

The theorem wants to find the probability of Y given that event X is true. We can therefore call X the evidence. P(Y) is the probability of Y happening in general without evidence. If we have the evidence, we get P(Y|X) or the likelihood of the

event after X is given. For our classification case, we can think of Y as the class and X as features contributing to estimating the likelihood of Y being a class.

2.3.4 AdaBoost

AdaBoost is a meta-learning method created to increase the efficiency of binary classifiers. An Adaboost classifier tries a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but changes the weights of incorrectly classified instances. The following classifiers focus more on the problematic cases.

2.4 Transformer based neural networks

In this section, we will discuss transformers, their predecessor recurrent neural networks, and the background for transformer-based neural networks that we will use in this thesis.

2.4.1 Recurrent Neural Network

Transformers are a type of neural network architecture. In terms of analyzing complex data types such as images and text, neural networks are very effective. However, different types of neural networks are optimized for different types of data. In the years preceding the introduction of Transformers in 2017, we utilized deep learning to understand text using a model based on a Recurrent Neural Network (RNN) (Schmidt 2019).

Consider the case where you would like to translate a sentence from Norwegian into English. An RNN would take as its input an English sentence, process each word individually, and then, sequentially, produce its Norwegian counterpart. The order of words matters in language, and you cannot just shuffle them around. That's why the key word here is sequential (Dyer et al. 2016). "Ola bit the dog" means something very different from the sentence: "The dog bit Ola", even though the words are identical.

Consequently, any model that attempts to understand language must be able to capture word order, and recurrent neural networks accomplish this by processing one word at a time, in sequence (Zaremba et al. 2014). However, RNNs were not without their shortcomings. In the first instance, they had difficulty handling large text sequences, such as long paragraphs or essays. They often forget what happened at the beginning of a paragraph by the time they get to the end (Li et al. 2015). For example, an RNN-based translation model might have difficulty remembering the gender of the subject of a long paragraph.

Furthermore, RNNs are challenging to train. Known for being susceptible to the

vanishing/exploding gradient problem, they were notoriously difficult to construct (Dieng et al. 2016). Moreover, RNNs were difficult to parallelize since they processed words sequentially (Ouyang et al. 2017). As a result of this parallelization, you could not simply speed up training by using more GPUs, so in turn, you could not train them on a significant amount of data.

2.4.2 Transformers

Google's introduction of Transformers in its paper from Vaswani et al. 2017 resulted in a significant change. Transformer models are neural networks that learn based on the context in which they are used. Transformers learn by following the relationship between sequential data, such as the words in this paragraph. This is accomplished by utilizing ever-evolving mathematical techniques. We refer to these techniques as self-attention. Transformers are able to determine even the slightest influence or dependence between data elements.

Transformers have been demonstrated to perform well in many different applications within the fields of natural language processing and computer vision. Consequently, transformer-based solutions are becoming increasingly popular due to their excellent results and ease of use.

In the days before transformers, it was necessary to train neural networks on large datasets of labelled data. Performing the training was very time-consuming, as can be seen in the section about recurrent neural networks 2.4.1. A transformer can use all available data by mathematically representing patterns between elements.

Transformers follow the usual neural network architecture composition consisting of an encoder and decoder block to process data. It is the extra addition to these blocks that make transformers so potent. Seen in figure 2.3.

A positional encoder tagging system is used to tag all data elements entering and leaving the network. Using these tags, attention units determine the relationship between each element. Attention allows the computers to recognize the same patterns as humans.

Encoder

The encoder consists of a stack of identical layers with two sublayers for each layer. A multi-head self-attention pooling mechanism is layer one, and a position-wise, fully connected feed-forward network is layer two. A query's value and key are all derived from the output of the previous encoder layer in the encoder self-attention layer.

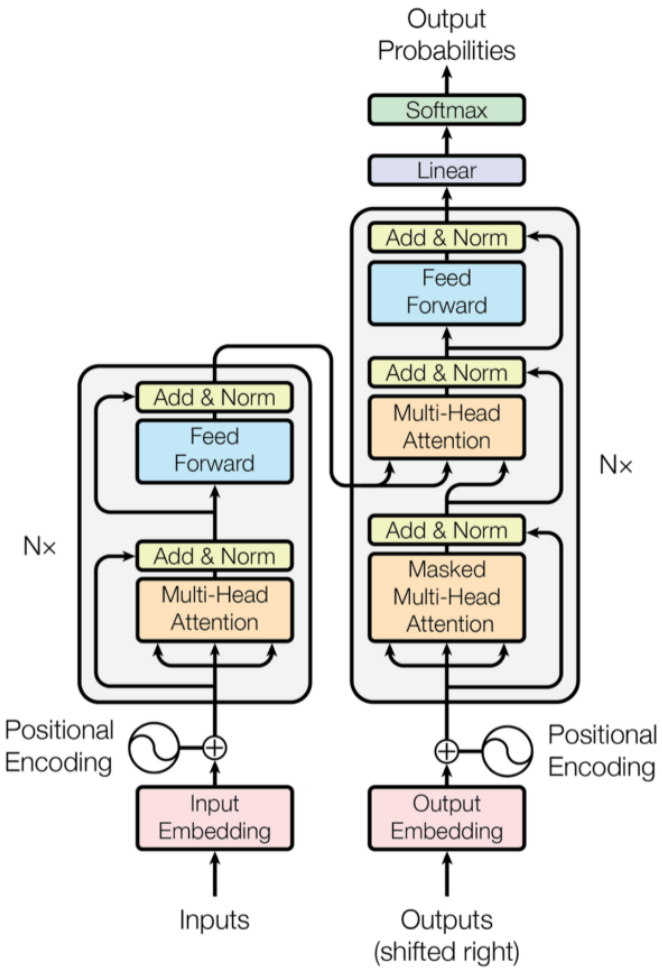


Figure 2.3: Transformer architecture (Taken from Vaswani et al. 2017 with permission from Illia Polosukhin).

Decoder

As with the encoder, the decoder consists of a stack of identical layers. In addition to the two sublayers in each encoded layer, the decoder has a third sublayer. The third layer performs multi-head attention on the encoder output.

Using this transformer, became the first sequence transduction model to be entirely based on attention, replacing the recurrent layers most commonly found in encoder-decoder architectures with multi-headed self-attention transformers.

2.4.3 BERT

Bidirectional Encoder Representation from Transformer (BERT) is a deep contextual language representation model. BERT was developed by researchers from Google in 2018 to pre-train bidirectional representations of words from unlabeled text. This is accomplished by jointly conditioning the left and right contexts in every layer. After this model is created, it can be fine-tuned by adding just one additional output layer to create exceptional performing models for various tasks.

By conditioning both left and right context simultaneously, BERT is designed to prepare deep bidirectional representations from an unlabeled text. Therefore, the pre-trained BERT model can be fine-tuned using just one additional output layer to create state-of-the-art models for a wide range of NLP tasks. It continues to learn unsupervised from the unlabeled text and improves even when used in practical applications (Chang 2018). Its pre-training serves as a base layer of "knowledge" to build from. From there, BERT can adapt to the ever-growing body of searchable content and queries and be perfected to a user's specifications.

Unlike directional models which read the input text sequentially (left-to-right or right-to-left), BERT reads the entire sequence of words simultaneously. It is therefore considered bidirectional. More precisely, it is non-directional. By being bidirectional, the model can learn the context of a word based on its surroundings.

BERT uses a method called masked language modeling to prevent the word in focus from "seeing itself", that is, having a fixed meaning independent of its context. BERT is forced to identify the masked word solely based on context. In BERT, words are defined by their surroundings, not by a pre-fixed identity.

In order to accommodate a wide range of NLP tasks, the BERT model can also receive both a single sentence and a pair of sentences as input. The sentences in this context are not necessarily linguistic sentences, but rather sequences of input tokens. You may use either a single sentence or two sentences. Thus, the work from Adhikari et al. 2019 is particularly interesting because it examines the use of BERTs on comprehensive documents rather than individual sentences, which is the focus of this thesis as well.

LEGAL-BERT

As previously discussed, BERT has demonstrated impressive results in several NLP tasks. A number of variations have developed from BERT's success, including Liu et al. 2019. However, only limited research has been conducted on its application to specialised domains. Chalkidis, Fergadiotis, Malakasiotis et al. 2020 has developed a version of BERT that focuses on the legal field, known as LEGAL-BERT.

Legal-BERT has been pre-trained on 12 GB of diverse legal English text. During the training, the algorithm learns more about legal text than the "regular" BERT (ibid.). It was found that Legal-BERT had marginal improvements in binary classification tasks with about 1% higher accuracy than BERT and 2.5% higher accuracy in multi-label classification on legal classification tasks on the ECHR dataset.

RoBERTa

RoBERTa stands for Robustly Optimized BERT-pretraining Approach and was introduced by Liu et al. 2019. RoBERTa builds upon BERT by effectively fine-tuning BERT with further training.RoBERTa is essentially BERT retrained over 160GB of additional text. Adding to the baseline of Wikipedia and corpus of books that BERT was trained on RoBERTa was further trained on CommonCrawl News Data, stories and text from OpenAI GPT (ibid.).

2.4.4 BigBird

BigBird was proposed by Zaheer et al. 2020 and is a sparse-attention based transformer that extends the usual Transformer based models like BERT for longer sequences.

BigBird uses sparse-attention but also global attention and random attention on the input sequence. This is due to the theory of using all three attention types; sparse, global and random attention, which equals full attention while being much more computationally efficient on longer sequences. BigBird is therefore an excellent tool for modeling long NLP tasks using a network.

2.4.5 ELECTRA

The ELECTRA model was introduced in K. Clark et al. 2020. ELECTRA is a new approach to pretraining that trains two transformer models. One is the "generator", and the other is the "discriminator". The generator replaces tokens in a sequence trained as a masked language model. The discriminator identifies which tokens the generator replaced in the series.

With ELECTRA, a small masked language model is trained together with the discriminator as the generator. The generator can be any model that produces an output distribution over tokens. In spite of the fact that the model resembles a generative adversarial network (GAN), the generator is trained with maximum likelihood rather than adversarially due to the difficulty of applying GANs to text. After pre-training, the generator is eliminated, and only the discriminator (the ELECTRA model) is fine-tuned on downstream tasks.

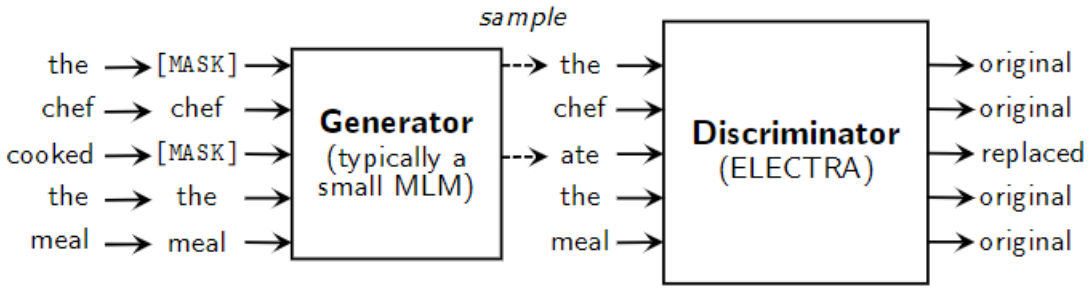


Figure 2.4: An overview of how replaced token detection works. (Used with permission from Kevin Clark (K. Clark et al. 2020))

2.4.6 XLnet

In Yang et al. 2019, the XLNet model is proposed. XLNet is a method for learning unsupervised language representations based on generalized permutation language models. In terms of longer text language tasks, XLNet has demonstrated excellent performance using Transformer-XL as its backbone model.

According to the paper from ibid., BERT corrupts input with masks that remove the dependency between the masked positions and suffers from a pretrain-finetune discrepancy. They propose a generalized autoregressive pretraining method in order to counteract this corruption. In particular, XLNet is able to learn bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order and overcome the limitations present in BERT through its autoregressive formulation (ibid.).

Chapter 3

Related work

In this chapter, we will examine methods that provide different approaches to our text classification problems, as well as work that has addressed similar issues. Additionally, we will examine the reasons why solutions such as ours are needed.

3.1 Transformers a foundation model

In the paper "On the Opportunities and Risks of Foundation Models" written by a large team from Stanford they refer to Transformers and BERT, particularly as a "foundation model". They name foundation models as models that are trained on a broad data at scale and are adaptable to a wide range of downstream tasks (Bommasani et al. 2021). They call them foundation models to emphasize their critically central yet incomplete character.

The team shows how transformer-based models are vastly powerful and central in deep learning yet have not reached their full potential. Many new variations of transformer models have been created, contributing to the ever-growing transformer-based model pool.

3.2 Neural Legal Judgment Prediction in English

Chalkidis, Androutsopoulos, and Aletras are the authors of the main paper upon which this thesis is based. Specifically, they were interested in predicting the outcome of a court case automatically, given the details of the case (Chalkidis, Androutsopoulos et al. 2019). For the first time ever, they applied neural networks to English legal judgment datasets in order to make predictions (ibid.). It appears that this started a trend within the natural language processing field in legal judgements as more papers have been published since, such as Chalkidis, Fergadiotis, Kotitsas et al. 2020 and Lage-Freitas et al. 2022.

Togheter they looked at a big variety of different neural models on binary classi-

fication, multi-label classification and case importance prediction (Chalkidis, Androutsopoulos et al. 2019). My work will continue on their binary and multi-label classification as they did not "deep-dive" into any specific models.

In general their best performing model was off their own creation. Chalkidis et al. proposed a hierarchical version of BERT that bypasses the length limitation of 512 wordpieces BERT has. This was one way to circumvent the problem BERT faces however, in this thesis I will look deeper into ways to solve BERTs length limitation and further improve upon Chalkidis et al. results.

3.3 Growing need for NLP solutions in the legal system

We will discuss different approaches for e-discovery and NLP solutions within the legal field in this section. Firstly, we will determine what parts will be relevant to this thesis. To begin with, we will examine how electronic discovery is currently conducted and how it came about.

Let us first make clear what "classic" discovery is.In a legal case, discovery is the process by which one party (the producing party) makes available to the other (the requesting party) any relevant materials that are contained in their possession. (Oard and Webber 2013). This Discovery phase was done with pen and paper documents, while nowadays it is mainly electronic discovery that is used.

In 2010, Conrad stated the need for artificial intelligence as information retrieval in e-discovery and as a whole (Conrad 2010). The industry of e-discovery has rapidly been growing since 2005. E-discovery market revenues are growing from over 1.8 $ billion in 2015 to over 3.7 $ billion by 2019—an average annual growth rate of 19% (RADICATI Group 2019). Just like the growth rate, the industry has seen different techniques and companies arise.

3.4 Document BERT

Adhikari et al. 2019 looked at what they called a "straightforward" classification model using BERT to achieve state-of-the-art results on four popular datasets. There are a few characteristics of document classification that might lead one to believe BERT is not the most appropriate model. For instance, syntactic structures matter less for content categories, and documents are typically much longer than BERT input. Consequently, the team optimized the BERT model by adding an additional soft-max classifier parameter, as shown in figure 3.1. For both single-label and multi-label tasks, they minimized the cross-entropy and binary cross-entropy loss.

$$W \in \mathbb{R}^{K*H} \tag{3.1}$$

The formula 3.1 is softmax classifier parameters for finetuning BERT for document classification. H represents the dimension of the hidden state vectors, and K represents the number of classes.

To alleviate the computational burden associated with BERT, the team applied knowledge distillation (Hinton et al. 2015) to transfer the knowledge from BERT to the smaller state-of-the-art model Bidirectional Long Short Term Memory.

3.5 Domain specific Natural Language Processing

The pretraining of large neural language models, such as BERT, has yielded impressive results in NLP (Dai et al. 2020). Nevertheless, most pretraining efforts focus on general domain corpora, such as Wikipedia. There is a general assumption that even domain-specific pretraining can benefit from starting with general-domain language models.

The article from Gu et al. 2021 challenges this assumption by showing that for domains like biomedicine, which have a large amount of unlabeled text, pretraining language models from scratch results in significant performance improvements over continuous pretraining of general-domain language models. These findings demonstrate that domain-specific pretraining contributes to developing new state-of-the-art results across a wide range of biomedical NLP tasks.

Researchers from the paper ibid. explored the problem of summarizing unstructured materials from scientific text in order to keep up with the constant flow of new literature being published. It may be possible to resolve this problem by using named entity recognition (NER) to extract structured summary-level data from unstructured materials science text.

Named entity recognition is the process of identifying and categorizing key information (entities) in text. Any word or set of words that refer consistently to the same thing can be considered an entity. Detected entities are classified into predetermined categories. A NER model may, for instance, detect the word "Buss" in a text and identify it as a "Car".

On three materials datasets[1], they compared the performance of four NER models. Four models are included, a bidirectional long short-term memory (BiLSTM) and three transformer models (BERT, SciBERT, and MatBERT) with varying levels of domain-specific materials science pre-training. MatBERT improves over the

[1]Source for the datasets `https://figshare.com/articles/dataset/NER_Datasets_DOIs_and_Entities_Doping_and_AuNP_/16864357`.

other two BERTBASE-based models by 1% to 12%, implying that domain-specific pre-training provides measurable advantages. BiLSTM consistently outperformed BERT despite its relative simplicity, perhaps due to its domain-specific pretrained word embeddings(Gu et al. 2021).

As a result, the researchers hypothesized that the measurable advantages previously demonstrated with domain-specific pre-training could be applied to models specific to narrower scientific disciplines such as materials science. The team concluded that domain-specific pre-training for large transformer models is still an open question in the field of NLP domainspecific.

Chapter 4

Methods and Implementation

Classifying large text datasets is a challenging task when using transformer-based algorithms. To aid our algorithms, we want high-quality data. It is essential for training neural networks and machine learning techniques. This means we need a good-sized dataset that does not have too significant an imbalance. This chapter will look at choosing a suitable dataset for training our algorithms and the general architecture structure they will follow.

4.1 Dataset

4.1.1 Choice of dataset

The European Court of Human Rights (ECHR) dataset published by Chalkidis, Androutsopoulos et al. 2019 bases itself on allegations of breaches of the human rights provisions. The dataset comes in two variations. The first is unaltered, while the other has anonymized identifiers of who and where the alleged violated article took place.

We chose to use the unaltered version of the dataset due to the anonymized version of the dataset showed little effect on classification results (ibid.) and because biases in algorithms is not the focus of this thesis.

4.1.2 ECHR dataset

The ECHR dataset consists of 11,500 cases from the ECHR's public database. Every case has a text describing the facts of the case. However, there are several other data features to consider:

BRANCH: Which branch of the court the case was held in. Going from admissibility Court -> Committee -> Chamber -> Grandchamber.

DATE: Which year the case is from.

IMPORTANCE: How important the ECHR considered the case ranging from a 1 representing a critical case contributing to the development of case law and a 4 being unimportant.

RESPONDENT: Which country is accused of the human rights article breach.

VIOLATED ARTICLES: Which article was violated (the classification target).

According to figure 4.1, the cases that are of greater importance than four often involve breaches of human rights article(s) especially that cases with importance score of three.It will be interesting to observe what impact the addition of such information will have on the classification.

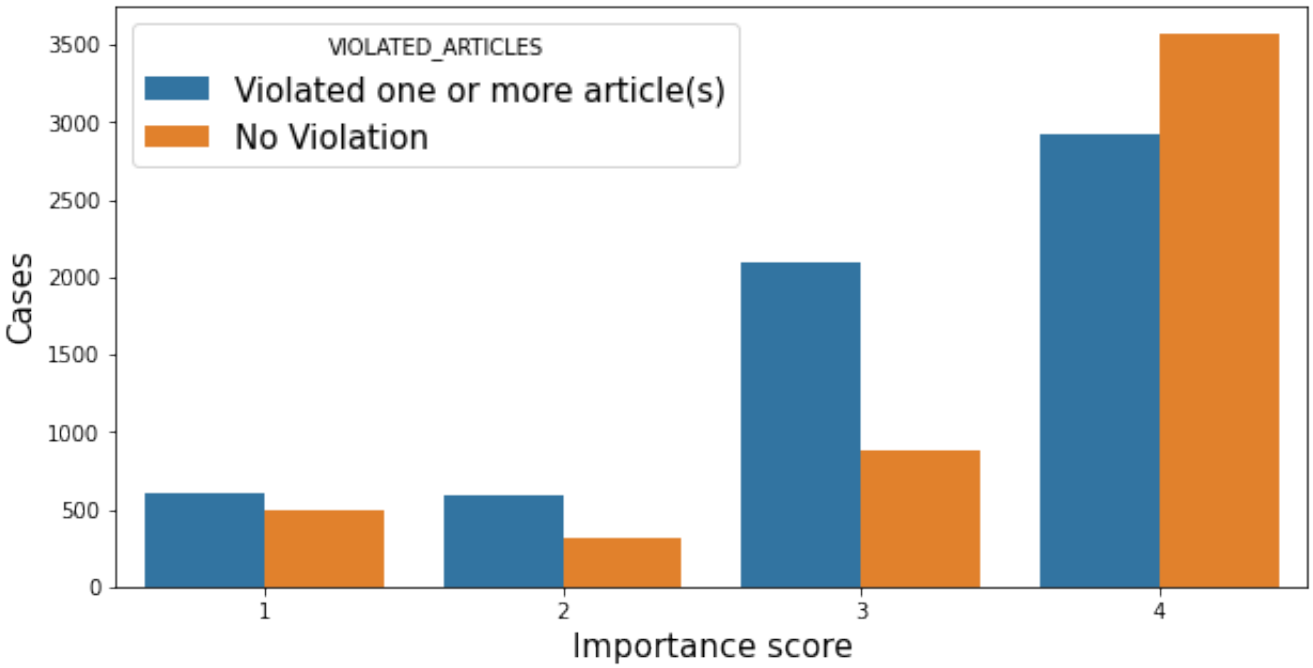


Figure 4.1: Overview

The case details range in length from 317971 to 68 characters, with an average of 13585 characters.

4.1.3 Balancing the dataset

A big part of the quality of a dataset is how balanced the dataset is. The distribution of violated articles in the dataset can be seen in the table below. There are 5810 cases of no violated articles and 5668 instances where a human rights article was violated. This is a very similar distribution between the binary case of

Violated Article / Protocol	***Occurence***
No Violation	5263
Violated Article 6: Right to a fair trial	3055
Violated Article 3: Prohibition of torture	1170
Violated Article 5: Right to liberty and security	1109
Violated Article 13: Right to an effective remedy	933
Violated Article 8: Right to respect for private and family life	734
Violated Protocol No. 1:	547
Violated Article 2: Right to life	382
Violated Article 10: Freedom of expression	355
Violated Article 14: Prohibition of discrimination	161
Violated Article 11: Freedom of assembly and association	143
Violated Article 34: Individual applications	82
Violated Article 9: Freedom of thought, conscience and religion	60
Violated Article 38: Individual applications	36
Violated Article 7: No punishment without law	27
Violated Protocol No. 4:	23
Violated Protocol No. 7:	23
Violated Article 4: Prohibition of slavery and forced labour	9
Violated Article 12: Right to marry	9
Violated Article 18: Limitation on use of restrictions on rights	9
Violated Article 25: Plenary Court	5
Violated Protocol No. 12:	3
Violated Protocol No. 6:	2
Violated Article 46: Binding force and execution of judgments	1

Table 4.1: Violation overview for the ECHR dataset before scraping.

classifying if an article is violated or not.

There is a big gap between the most violated human rights article: Article 6: Right to a fair trial with 3055 cases and Protocol No. 1 with only a single case.

To balance the dataset, we considered both scraping and data augmentation.However, due to the low number of cases ranging from one to nine, we did not have a large sample size for data augmentation. If we changed too few words, the newly generated sentences wouldn't have sufficient differences from the original ones to be distinguished by the classification model, which is an insignificant data augmentation outcome. It is also possible that the original sentences will be transformed into entirely different ones if too many changes are made, which could result in the loss of information that contributes to the classification of the cases (Gao 2020). We, therefore, went with data scraping to collect more data from the public ECHR database.

4.1.4 Scraped dataset

To collect more data, we scraped the ECHR public database[1] for new cases. The points of the case scraper is:

1. **Initialize scraper** Go to database overview page.
2. **Load all cases** Loads the full webpage by scrolling to the bottom of the main page.
3. **For each case and for every accompanying case details do the following:**
    a. **Extract case name** Extract what the case name. E.g: "In the case of X v. the Czech Republic".
    b. **Extract full article text** As it is not possible to only extract the facts about the case from the case overview, we extract all text about the case.
    c. **Go to case details** Go deeper into the specific case by going to the case details page.
    d. **Extract importance level** Extract what importance level the case was evaluated to be. This ranges from one to four.
    e. **Extract originating body** Says which court the case was judged in. E.g.: "Grand Chamber".
    f. **Extract article(s) broken** Exactly what articles where broken. E.g: "8, P1" for article eight and protocol number one.
    g. **Extract case language** Extract what language(s) the case is written up in.

The flow of the scraper can be seen in figure 5.1. The figure shows how the scraper moves for every case that is scraped. Starting from the main overview page from the database that shows every case, our search is specified. After that, it loops for every case article found and extracts the necessary data from that page. Finally, coming to the end of the iteration in the case details page where the scraper extracts the final information needed.

A problem encountered when scraping the ECHR database is how it was impossible to specify just the facts needed for extraction. Because we had to extract the whole case article, we needed to use regular expressions (regex) to remove unnecessary text. Removing all texts before "THE FACTS" and keeping all text until the next section titled "RELEVANT LEGAL FRAMEWORK AND PRACTICE."

[1] ECHR public database: `https://hudoc.echr.coe.int/eng`.

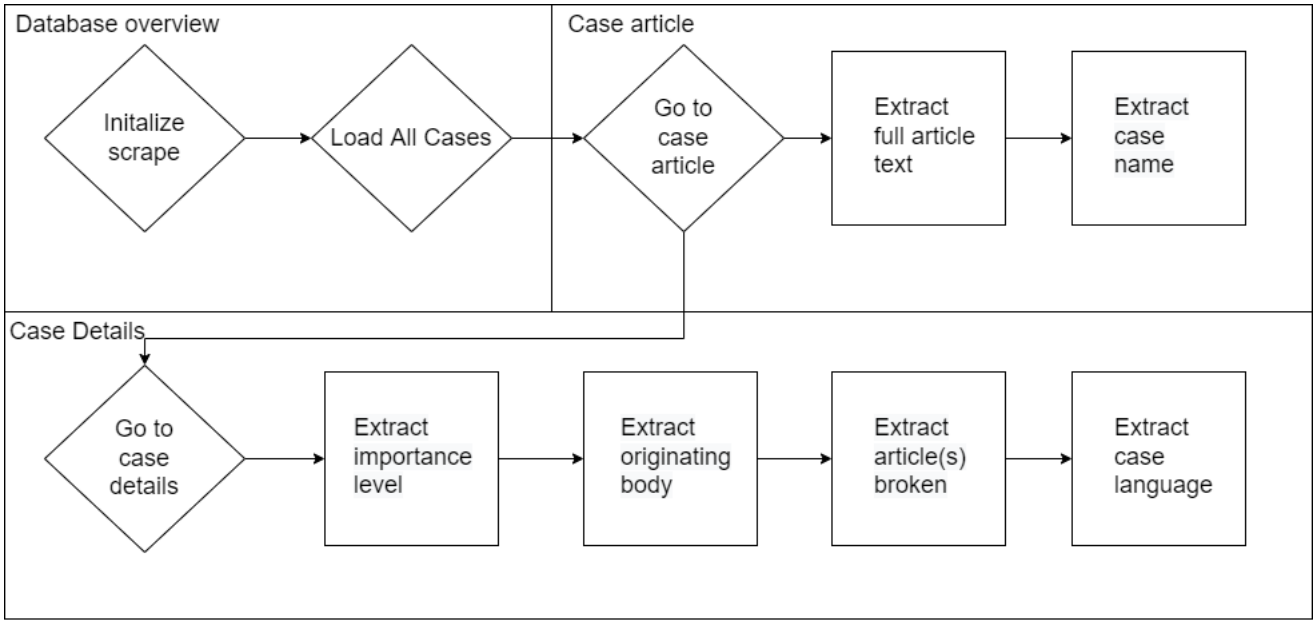


Figure 4.2: Scrape flow

The end result of scraping is that the new dataset only has three instead of seven violations with under twenty occurrences. Table 4.2 shows the total amount added to each class label. With a total sum of 1979 new occurrences of violated articles across 1105 new cases.

Violated Article / Protocol	*Occurence*
No Violation	5263 (+0)
Violated Article 6: Right to a fair trial	3502 (+ 447)
Violated Protocol No. 1:	1520 (+ 973)
Violated Article 3: Prohibition of torture	1212 (+ 42)
Violated Article 5: Right to liberty and security	1175 (+ 66)
Violated Article 13: Right to an effective remedy	1039 (+ 106)
Violated Article 8: Right to respect for private and family life	772 (+ 38)
Violated Article 2: Right to life	393 (+ 11)
Violated Article 10: Freedom of expression	370 (+ 15)
Violated Article 14: Prohibition of discrimination	178 (+ 17)
Violated Article 11: Freedom of assembly and association	151 (+ 8)
Violated Article 34: Individual applications	107 (+ 25)
Violated Protocol No. 7:	83 (+ 60)
Violated Article 9: Freedom of thought, conscience and religion	71 (+ 11)
Violated Article 7: No punishment without law	57 (+ 30)
Violated Article 46: Binding force and execution of judgments	52 (+ 51)
Violated Article 18: Limitation on use of restrictions on rights	51 (+ 42)
Violated Article 38: Examination of the case	36 (+ 0)
Violated Protocol No. 4:	29 (+ 6)
Violated Article 4: Prohibition of slavery and forced labour	20 (+ 11)
Violated Protocol No. 12:	20 (+ 17)
Violated Article 12: Right to marry	9 (+ 0)
Violated Article 25: Plenary Court	5 (+ 0)
Violated Protocol No. 6:	5 (+ 3)

Table 4.2: Violation overview for the ECHR dataset after scraping. Increase in occurrence is shown in parenthesis

4.2 Architecture

We will train various models to classify the datasets. A complete list of the exact architecture for each model can be found in table 4.3.

All models except LEGAL-BERT are pretrained on a joint base dataset of the BooksCorpus and English Wikipedia, totalling 13GB of plain text. However, most have some deviation in what dataset they are trained on.

LEGAL-BERT differs from other models in that it is developed exclusively from domain-specific data. The training data includes EU legislation, UK legislation, European Court of Justice decisions, US court cases, US contracts, and data from the ECHR database. The total amount of legal data used for pretraining LEGAL-BERT totals 12 GB. Interestingly, LEGAL-BERT has been pretrained on data from

Model	Dataset pretrained on
bert-base-uncased	Base
legal-bert-base-uncased	Legal corpora
roberta-base	Base, cc-news, openwebtext, stories
bigbird-roberta-base	Base, cc-news
electra-base-discriminator	Base, Giga5, ClueWeb,Common Crawl
xlnet-base-cased	Base, Giga5, ClueWeb,Common Crawl

Table 4.3: Overview of what models used for each neural network and what datasets they were trained on. "Base" refers to BooksCorpus and English Wikipedia

the ECHR dataset before being trained on the same dataset for classification tasks in this thesis.

ELECTRA is trained on the same dataset as XLnet comprised of BooksCorpus, English Wikipedia, Giga5, ClueWeb 2012-B (19GB) and Common Crawl (110GB) (Yang et al. 2019).

In comparison to the other models we test, BigBird can handle a maximum sequence length of 4096, which is 8x the maximum sequence length of 512 that the other models have. The specific parameters that change before training each model are:

- What model to be used.

- Exactly which architecture from the model to be used. Using architectures from Huggingface seen in 4.3.
- Which features to be added (We will use both importance and branch).

- Maximum sequence length. For most of the models the maximum sequence length is 512 with the exception of BigBird which has a maximum sequence length of 4096.

- What extra facts to be added to the facts about the case. This can be what branch of court the case heard, what importance score the case was given or both.

Chapter 5

Experiments and Results

This chapter describes the experimental set-up used for the experiments presented in the previous chapters and the results obtained from the experiments. First, we present the experimental set-up.

5.1 Experimental setup

In this section, we look at the experimental setup. Presenting both the method of sampling used on the dataset, what hardware was used and the hyperparameter and metrics used.

5.1.1 Hardware

The experiments were performed on the same computer.The computer was kept off-line for this period and had only essential software running in the background. Both training and evaluations were run on an NVIDIA GeForce GTX 1060 GPU with 6GB of RAM. This allowed for experimentation on larger models when using a low batch size.

5.1.2 Dataset

For research confidentiality purposes and to later achieve the reproducibility of the results, we will use publicly available data from the ECHR database. We are going to use both the datasets presented from section 4.1 as the basis for training and evaluating multi-classification models and only the original base dataset for binary classification.

The split between Violated and Non-Violated articles can be seen in table 5.1. As there was no a increase in the number of non-violated cases in the base dataset will we will only do binary tasks on the orgininal dataset. For the experiments, the dataset used had all their instances labelled. The datasets will always be divided into a 70/30 split for training and evaluation.

Dataset	Violated	Non-Violated
Base dataset:	6215	5263
Scraped dataset:	7320	5263

Table 5.1: Case split between Violated and Non-Violated. Note that each case can have multiple violated articles, but only count as a single violated case

5.1.3 Data Preparation

Transformer models typically restrict the maximum length allowed for a sequence. The length is defined as the number of tokens, where a token is any of the "words" that appear in the model vocabulary. Unfortunately, each model type also has an upper bound for the token length, most commonly 512.

Padding and truncation are strategies for dealing with this problem. The padding adds a special padding token to ensure shorter sequences will have the same length as either the longest sequence in a batch or the maximum size accepted by the model. Truncation works in the other direction by truncating long sequences. Truncation shortens long input text to fit the maximum length size set by models. We want to avoid this as we want as much text as possible for training our models.

While there is currently no standard method of circumventing this issue, a plausible strategy is to use the sliding window approach. Any sequence exceeding the maximum sequence length will be split into several windows (sub-sequences). However, doing so will increase the training time of the models as all available text will now be trained on rather than shortened down from truncation.

The windows will overlap to a certain degree to minimize any information loss

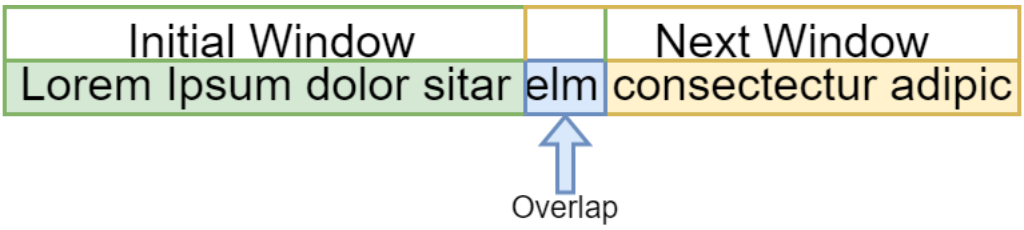


Figure 5.1: Sliding window example in a smaller scale than the larger 512 sequence windows

that hard cutoffs may cause. The amount of overlap between the windows is determined by the stride. The stride is the distance in terms of the number of tokens that the window will be moved to obtain the next sub-sequence. We will set the stride to 0.9 of the maximum sequence length resulting in about 10% overlap between the sub-sequences.

Another problem from using a sliding window is that the model has to classify several sub-sequences of the entire input text depending on the maximum length available to the model and the amount of input case text. The multiple classifications needed on all sub-sequences will further increase the training time. The total number of training samples will also grow to be higher than the number of sequences originally in the train data.

5.2 Experiment 1: Binary classification

In this section we are focusing on the binary classification. The dataset has two classes where each case has to belong to one of those classes. We will also look at the results of adding additional features to the dataset. Every document present in the datasets is labelled with one class only when doing binary classification.

Due to the large size of the raw text in the dataset, we had to train on a very small batch size of four. In combination with the small batch size, we trained the algorithms at a learning rate of 4e-5 over ten epochs. This amount of epochs was chosen as a balance between training time for every model and performance.

5.2.1 Results

Comparing the results from our models seen in table 5.2 to the original paper from Chalkidis, Androutsopoulos et al. 2019 which best performing model was HIER-BERT a BERT model that allowed for a bigger length sequence which had an F1 score of 82. We can see there has already been a big improvement.

The improvements are likely due to the stride applied to allow for learning bey-

Model	Precision	Recall	F1-Score
BERT	86.6	85.6	85.9
LEGAL-BERT	86.3	85.4	85.6
RoBERTa	**89.1**	**86.2**	**86.7**
BigBird	86.9	86.1	86.3
ELECTRA	86.1	85.1	85.3
XLNet	86.7	86.0	86.2

Table 5.2: Results from neural network models without any added features

ond the transformer's usual limited length. As ibid. had an abysmal performance from BERT with an F1 score of 17.0. The F1 score is worse than just randomly guessing. The likely culprit is the truncation of the case facts BERT was trained on

Model	Precision	Recall	F1-Score
COIN-TOSS	50.4±0.7	50.5 ±0.8	49.1 ±0.0
BOW-SVM	71.6±0.0	72.0 ±0.0	87.8 ±0.0
BERT	24.0±0.2	50.0 ±0.0	17.0 ±0.5
HIER-BERT	**90.4**±0.3	**79.3** ±0.9	**82.0** ±0.9
HAN	88.2±0.4	78.3 ±0.2	80.5 ±0.2
BIGRU-ATT	87.0±1.0	77.2 ±3.4	79.5 ±2.7

Table 5.3: Results from Chalkidis, Androutsopoulos et al. 2019

(Chalkidis, Androutsopoulos et al. 2019). As in our case, using a striding window allowed for BERT to train on the whole case description, which in turn resulted in an F1 score of 85.9.

The best performing model on the base dataset without any added features was RoBERTa. This is not surprising as RoBERTa has been modified in essence to improve on BERT. The modifications were done by increasing the amount of pre-training data and hyperparameter tuning.

Model	Precision	Recall	F1-Score
SVM	80.0	79.4	79.6
Decision Tree	78.6	78.5	78.5
Naïve Bayes	71.4	69.5	69.4
Ada boost	**81.9**	**81.5**	**81.7**

Table 5.4: Results from conventional machine learning techniques without any added features.

Using the results from multiple conventional machine learning techniques shown in table 5.4 as a comparison for the transformer-based models, it is clear that neural models are superior.

Class	Precision	Recall	F1-score	Cases
No Violation:	96.3	75.0	84.3	1597
Violated:	81.9	97.5	89.0	1847

Table 5.5: Classwise performance from the best performing model RoBERTa.

5.3 Experiment 2: Multi classification

Multi-classification is the subject of this section. In multi-classification, the labels are typically more similar than the binary task of a violation versus a non-violation. It is also assumed that categorizing into which of the twenty-three articles are violated is more complex than classifying if a human rights article has been violated.

5.3.1 Results

To be able to directly compare our results with the results from ibid., we trained the models on the original dataset and included no violation as a possible label. After scraping additional data, we retrained all the models on the new dataset.

Model	Precision	Recall	F1-Score
BERT	77.4	74.2	75.2
LEGAL BERT	78.4	74.5	76.1
RoBERTa	78.9	77.0	77.7
Bigbird	**80.0**	77.1	**78.1**
Electra	77.1	74.3	75.2
XLnet	79.3	**77.7**	77.9

Table 5.6: Results from multi classification with additional scraped data showing the weighted average results for each model.

The results are shown in Table 5.6. With a weighted F1-Score of 78.1, BigBird had the best results. It outperformed other models that have lower maximum sequence lengths.

Having fewer sub-sequences is believed to be the reason for BigBird's superior performance. Sub-sequences did not pose a significant problem for the other models in the binary classification task. As in the binary classification task, classifying each sub-sequence as either a violated or non-violated article resulted in the final full sequence classification only being affected by the sum of these two labels.

In the case of multi-label classification, however, an example of a single case with a sequence length of 5000 will result in ten sub-sequences. Within each sub-sequence there can be 23 different "sub-classifications" on which labels are violated. With a maximum sequence length of 4096, there would only be two subs equences for BigBird.

Compared to the previous unscraped dataset, BigBird and most of the other models had a much higher true positive result on the lower violated article counts. A diagram of the confusion matrix is presented in Figure 5.2.

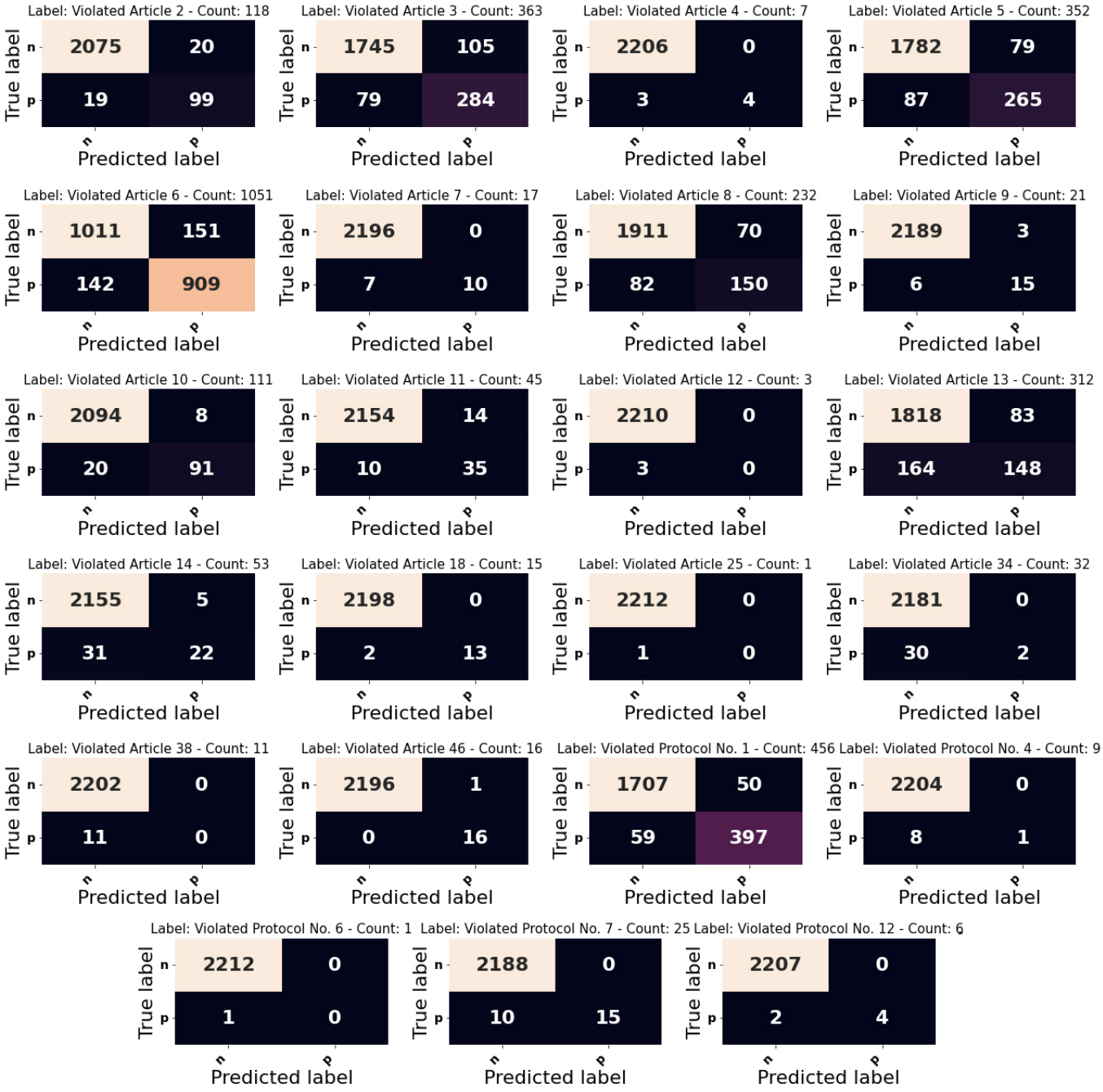


Figure 5.2: The figure shows the individual confusion matrices for each label based on the BigBird model's results.

As a result, some of the lower values, such as Article 18 and Protocol 12, had over 50% of their cases categorized correctly, even though they had only 15 and 6 cases in the evaluation set, respectively. Article 46 correctly predicted all 16 of its cases.

The confusion matrix for each individual label prediction from the best performing multiclassificaton model BigBird can be viewed in figure 5.2. Please note that the figure is an inverted version of the confusion matrix described in section 2.1.1. In this case, the TN is in position (0,0) and the TP is in position (1,1).

Chapter 6

Discussion

This chapter aims to discuss the results of the experiments presented in the previous chapter. We will perform a more in-depth quantitative examination of some of the results and examine the limitations of our work. Our discussion will conclude with a discussion of the research questions.

6.1 Further Analysis

RoBERTa produced the greatest result on the binary task, and BigBird performed the best on the multi-label classification task. We believe this shows the weakness of using a sliding window when multiple classification options exist. As the options become so numerous, it becomes increasingly difficult for the models to correctly classify all the generated subsequences, which led to BigBird, which had fewer subsequences, outperforming the others.

Despite the shortcomings outlined above, the addition of a sliding window had a positive effect. However, the addition of additional facts to the sequences yielded little to no results. The little impact is most likely due to the already large text base to classify from, and we would probably have seen a more significant effect of the additional facts on smaller text sequences.

6.2 Limitation of this work

Some parts of this thesis need further discussion. Some elements of this thesis had limitations, and in other circumstances, we would like to explain the reasoning behind the decisions we made.

Dataset

In order to perform text classification, we decided to use only one source for our dataset. The European Court of Human Rights has an extensive public database

previously used in text classification literature and is easily accessible. In addition to allowing future replication, the database allows the comparison of results from this thesis with those from other researchers.

Unfortunately, there are few examples of ECHR-based text classification applications to compare. Because this thesis scraped additional data for the multi-label classification task, the results may differ from those of others who use the base dataset from Chalkidis, Androutsopoulos et al. 2019. In addition, data augmentation could have been used to further balance out the dataset after scraping. This would have been a feasible option following scraping. However, due to the amount of time required to train the models on the already extensive input data as well as the amount of additional work required for data augmentation, this wasn't done.

Runtime

The runtime required to train the models was extensive, as this thesis used a variety of different models for classification of large datasets using time-consuming methods. Therefore, the possibility of playing around with different variations of hyperparameters was not available. Several parameters, such as learning rates, weights, and epochs, could have been tuned for better results.

6.3 Research Questions

RQ 1 - How effective are transformers in handling long sequences of text data from legal documents?

We found that using a sliding window approach to handle the long data sequences resulted in state-of-the-art results. Its downside is the large computational load and the potential for overfitting in the overlap between the steps.

RQ 2 - Which features can be exploited to train transformers and effectively classify legal documents?

We show through experiments that the addition of what should be impactful features had little effect due to the length of information already available to the models. However, we surmise that the impact would be more significant on smaller text sizes. Also, a possible ensembling model which could use both the transformers on the full-length text and the individual extra feature facts in another model would have better results.

RQ 3 - How viable would such a solution be in enhancing the efficiency of legal assistance?

We show the possibility of classifying cases not only as violated or not but specifically which ones are possible. Accordingly, we conclude that using models such as

the ones used in this thesis can help to reduce the cost of legal aid.

Chapter 7

Conclusion and Future Work

This thesis has focused on a transformer-based approach to legal text classification tasks. Text classification and transformers are growing research fields and are increasingly focused on by professionals. This thesis has looked at handling larger input text classification tasks that surpass the normal maximum length allowed by transformers.

7.1 Conclusion

We have completed experiments on six different neural models on two text classification. Within the text classification tasks we have looked at the results of adding additional data features to the initial data sequence resulting in various different workflows. According to these experiments, adding more information to the extensive text documents had little impact on the classification outcome.

As the amount of text data available to the models was already so large, there was little impact expected. Nevertheless, we expected at least some effect, because looking at the dataset, it was possible to draw some conclusions based on the importance given and from what branch cases came.

Moreover, the solution of a sliding window over the input text resulted in state-of-the-art results from various models. As the primary contribution of our paper, we present improved baselines that can form the basis for future research.

7.2 Future Work

While state-of-the-art transformer models have shown success on large legal text classification tasks, there is still room for improvement, as indicated by our experiments. In addition to the multi-label tasks, where the models are far from perfect, there is also much room for improvement in the binary classification tasks. In this section, we suggest potential areas for further improvement of the results in this thesis.

Domain specific pretraining

A potential model that could achieve good results in binary and multi-class classification could be a combination of Legal-BERT and Big Bird. Increasing the maximum sequence length, such as Big Bird, can help reduce the amount of overlapping and overfitting issues that may arise due to the sliding window technique. This new model could also be enhanced by being pretrained on a domain specific corpora, such as Legal BERT.

Ensemble model

Another possible model would be to create a transformer ensemble model that would emphasize the extra features we tried to introduce. It is possible that the model would allow for better use of these features than the transformer models and together make for a more successful classification model as a whole.

Assemble modelling could eliminate the noise of introducing new features to the long text. The transformer models currently rely too heavily on the case details and do not consider the additional features added.

Dataset Quality

Datasets are essential to the performance of transformer models. Obtaining high-quality datasets is a time-consuming process, and while we explored scraping additional data, further work exists to be done, and we propose that this be our focus. As a last note, there is exciting work being done on dataset augmentation, which could be further explored, such as Shorten et al. 2021 and Karras et al. 2020.

Bibliography

Adhikari, Ashutosh, Achyudh Ram, Raphael Tang and Jimmy Lin (2019). 'Docbert: Bert for document classification'. In: *arXiv preprint arXiv:1904.08398*.

Bommasani, Rishi et al. (2021). *On the Opportunities and Risks of Foundation Models*. DOI: 10.48550/ARXIV.2108.07258. URL: https://arxiv.org/abs/2108.07258.

Buckland, Michael and Fredric Gey (1994). 'The relationship between recall and precision'. In: *Journal of the American society for information science* 45.1, pp. 12–19.

Carlin, Jerome E and Jan Howard (1964). 'Legal representation and class justice'. In: *UCLA L. Rev.* 12, p. 381.

Chalkidis, Ilias, Ion Androutsopoulos and Nikolaos Aletras (2019). *Neural Legal Judgment Prediction in English*. DOI: 10.48550/ARXIV.1906.02059. URL: https://arxiv.org/abs/1906.02059.

Chalkidis, Ilias, Manos Fergadiotis, Sotiris Kotitsas, Prodromos Malakasiotis, Nikolaos Aletras and Ion Androutsopoulos (2020). 'An empirical study on large-scale multi-label text classification including few and zero-shot labels'. In: *arXiv preprint arXiv:2010.01653*.

Chalkidis, Ilias, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras and Ion Androutsopoulos (2020). *LEGAL-BERT: The Muppets straight out of Law School*. DOI: 10.48550/ARXIV.2010.02559. URL: https://arxiv.org/abs/2010.02559.

Chang, Author Ming-Wei (Oct. 2018). 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'. In: 1.1, pp. 1–16.

Chaturvedi, Ajay Kant and RL Koul (2019). 'Legal Aid and Legislative Initiatives'. In: *Think India Journal* 22.14, pp. 11256–11266.

Clark, Kevin, Minh-Thang Luong, Quoc V. Le and Christopher D. Manning (2020). *ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators*. DOI: 10.48550/ARXIV.2003.10555. URL: https://arxiv.org/abs/2003.10555.

Clark, Peter and Tim Niblett (1989). 'The CN2 induction algorithm'. In: *Machine learning* 3.4, pp. 261–283.

Conrad, Jack G (2010). 'E-Discovery revisited: the need for artificial intelligence beyond information retrieval'. In: *Artificial Intelligence and Law* 18.4, pp. 321–345.

Dai, Xiang, Sarvnaz Karimi, Ben Hachey and Cecile Paris (2020). 'Cost-effective selection of pretraining data: A case study of pretraining BERT on social media'. In: *arXiv preprint arXiv:2010.01150*.

Davis, Jesse and Mark Goadrich (2006). 'The relationship between Precision-Recall and ROC curves'. In: *Proceedings of the 23rd international conference on Machine learning*, pp. 233–240.

Dieng, Adji B, Chong Wang, Jianfeng Gao and John Paisley (2016). 'Topicrnn: A recurrent neural network with long-range semantic dependency'. In: *arXiv preprint arXiv:1611.01702*.

Dyer, Chris, Adhiguna Kuncoro, Miguel Ballesteros and Noah A Smith (2016). 'Recurrent neural network grammars'. In: *arXiv preprint arXiv:1602.07776*.

Gao, Jie (2020). *Data Augmentation in Solving Data Imbalance Problems*.

Gu, Yu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao and Hoifung Poon (2021). 'Domain-specific language model pretraining for biomedical natural language processing'. In: *ACM Transactions on Computing for Healthcare (HEALTH)* 3.1, pp. 1–23.

Hinton, Geoffrey, Oriol Vinyals and Jeff Dean (2015). *Distilling the Knowledge in a Neural Network*. DOI: 10.48550/ARXIV.1503.02531. URL: https://arxiv.org/abs/1503.02531.

Hirsch, Donald (2018). 'Priced out of Justice? Means testing legal aid and making ends meet'. In.

— (2020). 'Establishing a national standard: the role of the UK's Minimum Income Standard in policy and practice'. In: *Minimum Income Standards and Reference Budgets*. Policy Press, pp. 307–318.

Jabbar, H and Rafiqul Zaman Khan (2015). 'Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study)'. In: *Computer Science, Communication and Instrumentation Devices* 70.

Karras, Tero, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen and Timo Aila (2020). *Training Generative Adversarial Networks with Limited Data*. DOI: 10.48550/ARXIV.2006.06676. URL: https://arxiv.org/abs/2006.06676.

Lage-Freitas, André, Héctor Allende-Cid, Orivaldo Santana and Lıvia Oliveira-Lage (2022). 'Predicting Brazilian court decisions'. In: *PeerJ Computer Science* 8, e904.

Li, Jiwei, Minh-Thang Luong and Dan Jurafsky (2015). 'A hierarchical neural autoencoder for paragraphs and documents'. In: *arXiv preprint arXiv:1506.01057*.

Liu, Yinhan et al. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. DOI: 10.48550/ARXIV.1907.11692. URL: https://arxiv.org/abs/1907.11692.

Oard, Douglas W and William Webber (2013). 'Information retrieval for e-discovery'. In: *Information Retrieval* 7.2-3, pp. 99–237.

Opitz, Juri and Sebastian Burst (2019). 'Macro f1 and macro f1'. In: *arXiv preprint arXiv:1911.03347*.

Ouyang, Peng, Shouyi Yin and Shaojun Wei (2017). 'A fast and power efficient architecture to parallelize LSTM based RNN for cognitive intelligence applications'. In: *Proceedings of the 54th Annual Design Automation Conference 2017*, pp. 1–6.

RADICATI Group, INC (2019). 'eDiscovery Market, 2015-2019'. In: pp. 3–4.

Raskutti, Garvesh, Martin J Wainwright and Bin Yu (2014). 'Early stopping and non-parametric regression: an optimal data-dependent stopping rule'. In: *The Journal of Machine Learning Research* 15.1, pp. 335–366.

Schmidt, Robin M (2019). 'Recurrent neural networks (rnns): A gentle introduction and overview'. In: *arXiv preprint arXiv:1912.05911*.

Shorten, Connor, Taghi M Khoshgoftaar and Borko Furht (2021). 'Text data augmentation for deep learning'. In: *Journal of big Data* 8.1, pp. 1–34.

Stortinget, Den særskilte komité nedsatt av (2015). 'Innstilling fra Den særskilte komité nedsatt av Stortinget 18. mars 2005 for behandling av St.meld. nr. 17 (2004-2005) Makt og demokrati'. In.

Tønnessen, Ingebjørg (2020). 'Likhet for loven — Lov om støtte til rettshjelp'. In: pp. 136–137.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin (2017). *Attention Is All You Need*. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.

Warde-Farley, David, Ian J Goodfellow, Aaron Courville and Yoshua Bengio (2013). 'An empirical analysis of dropout in piecewise linear networks'. In: *arXiv preprint arXiv:1312.6197*.

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov and Quoc V. Le (2019). *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. DOI: 10.48550/ARXIV.1906.08237. URL: https://arxiv.org/abs/1906.08237.

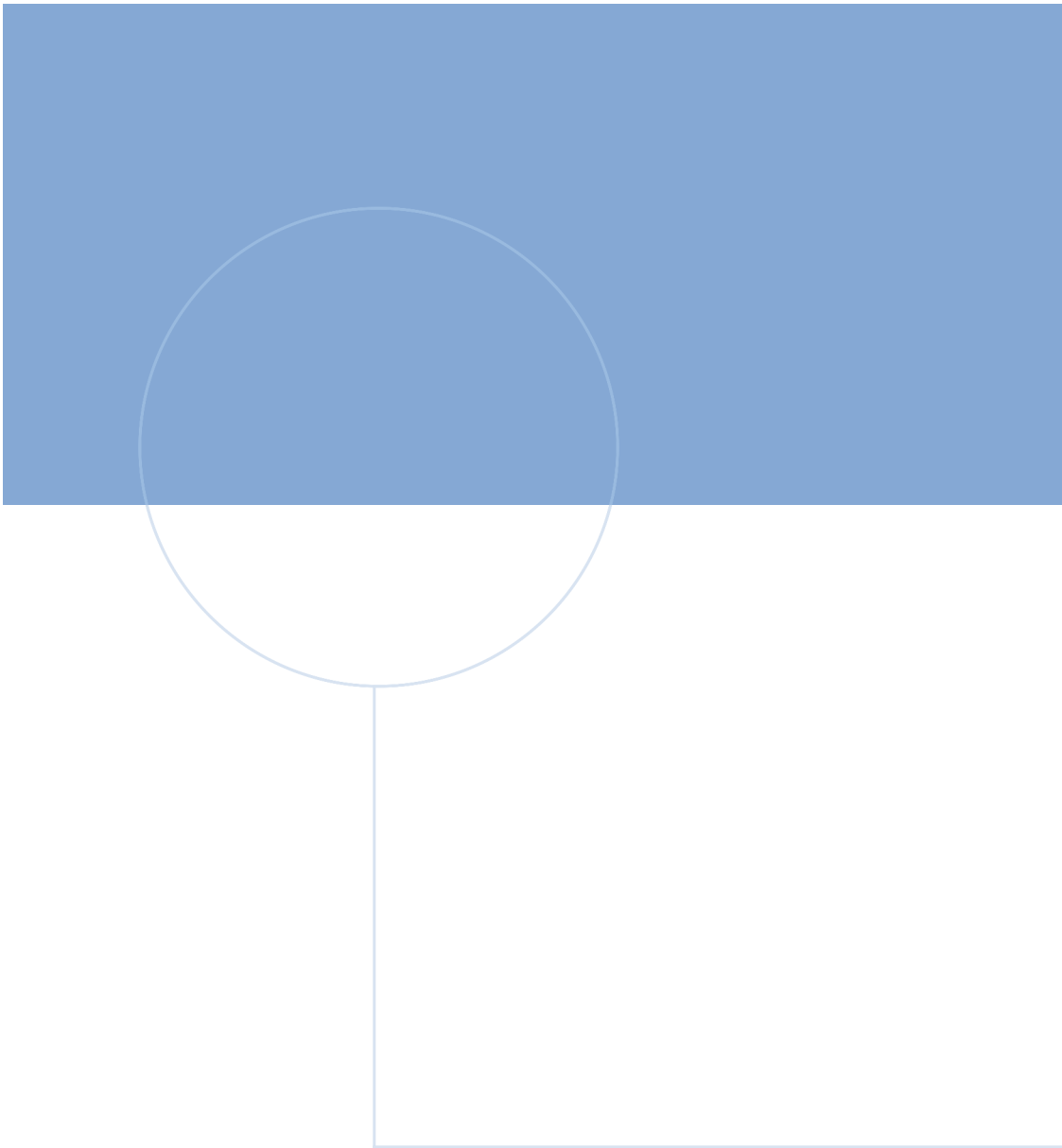
Yin, Ming, Jennifer Wortman Vaughan and Hanna Wallach (2019). 'Understanding the effect of accuracy on trust in machine learning models'. In: *Proceedings of the 2019 chi conference on human factors in computing systems*, pp. 1–12.

Ying, Xue (2019). 'An overview of overfitting and its solutions'. In: *Journal of Physics: Conference Series*. Vol. 1168. 2. IOP Publishing, p. 022022.

Yip, Kevin Y and Mark Gerstein (2009). 'Training set expansion: an approach to improving the reconstruction of biological networks from limited and uneven reliable interactions'. In: *Bioinformatics* 25.2, pp. 243–250.

Zaheer, Manzil et al. (2020). 'Big Bird: Transformers for Longer Sequences'. In: DOI: 10.48550/ARXIV.2007.14062. URL: https://arxiv.org/abs/2007.14062.

Zaremba, Wojciech, Ilya Sutskever and Oriol Vinyals (2014). 'Recurrent neural network regularization'. In: *arXiv preprint arXiv:1409.2329*.



NTNU

Kunnskap for en bedre verden