A Comparative Study Of Using Transformer Methods To Multi-Label Long Documents

Hanane el Aajati
12070955

Bachelor thesis
Credits: 18 EC

Bachelor *Kunstmatige Intelligentie*

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisors
Dr. R.G.F. (Radboud) Winkels

Faculty of Law
University of Amsterdam
Roetersstraat 11
1001 NB Amsterdam

Drs. (Roderick) W. Lucas

Vrije Univserieit Amsterdam
Faculty of Law
Manager (Deloitte, Netherlands)

Semester 2, 2022

Abstract

This research tries to find an efficient method to multi-label documents using Transformers. Transformers are powerful models because they are pre-trained on large amounts of data. A major drawback is that most transformers are not able to process documents longer than 512 tokens. We try to tackle this issue by proposing a new method to multi-label long documents - summarizing long documents first before multi-labelling with the bert-base-cased (bert) transformer. This summarization method is then compared with two already existing methods: truncating documents after the first 512 tokens and Longformer. The methods are evaluated on the F1 score and the results show that the Longformer performs the best. The summarization method and truncating method seem to output almost equal F1 scores. Although the summarization method did not perform well on the dataset used in this research, it could be promising for datasets with more structured documents.

Keywords: *Transformers, multi-label classification, Extractive summarization, Bert, Longformer*

Contents

Chapter 1

Introduction

One of the first steps taken by lawyers when receiving a legal matter to work on is finding similar cases. A method often used to find similar cases is by finding lawsuits labelled with labels that define the content of the case. To enhance organizational characteristics and encourage quick access to vital information, document collections are categorized according to their content using a set of labels generated from some type of thesaurus. For instance, most European parliaments utilize the Eurovoc thesaurus to classify legislative resolutions. The practice of assigning thesaurus labels to documents is typically done manually by a team of skilled documentalists. Automatically multi-labelling lawsuits could make the process of assigning labels to law cases an easier and more efficient task. This could increase these legal services' quality and efficiency and reduce the employee's workload.

One approach to automate the task of assigning labels to text documents is using text classification techniques. Text classification techniques are well-researched within the field of Artificial Intelligence (AI). Text classification includes methods such as binary classification[1], multi-class classification[2], and multi-label classification[3]. Unlike the first two methods in which the class labels are mutually exclusive, multi-label classification uses a specialized machine learning algorithm so that multiple labels can be assigned to one document. For example, a legal document could be about both human rights as well as tax law.

One of the most popular methods currently used for multi-labelling tasks are transformers [4]. The transformer architecture allows for effective parallel training and scales with training data and model size. The purpose of a transformer is to handle sequential input like textual data streams and it uses an attention mechanism to detect relationships between sequential elements. A powerful advantage of using a transformer is that they are able to capture patterns between sequential

elements in long ranges. For example, in the sentence “The old man always carries his car keys with him” the transformer would be able to catch the relation between ‘man’ and ‘him’. Even though these words have a longer distance between them. Recurrent neural networks for instance would not be able to capture this pattern and would have ‘forgotten’ that the sentence contained the word "man" when it reaches the word "him".

One of the fundamental problems with transformers is that the self-attention mechanisms grow quadratically with sequence length. This results in the model not being able to process longer sequences. The current maximum amount most transformer models are able to process is 512 tokens. This is a major drawback because real-world data does not limit itself to this maximum.

Currently, there is some research on how transformers perform on documents that have a maximum amount of 512 tokens. However, methods to use transformers on longer documents appear to be understudied. Some methods have been proposed by researchers to process longer documents. The most standard way is to truncate every token after the 512th token is processed. Meaning only the first 512 tokens are taken into account by the transformer model [5]. Other approaches that have been researched are transformers like Longformer [6] and Bird [7]. Although these types of transformers provide a way to process longer documents, they do have the disadvantage that they are expensive in their computing time. Another more recent approach is to minimize the document length by dividing the documents into chunks of 512 or less [8].

Although a few studies have been done to analyze the performance of transformers on longer documents, there is no study that tries to summarize the data first before applying a transformer model to it. This research will analyze the results of summarizing the data first before multi-labelling it with a transformer. We will then compare the results with the two already existing methods; Longformers and truncating the document length after 512 tokens. The method of truncating the document length after 512 tokens will be referred to as ‘truncating’ in the rest of this research and the method of summarizing before multi-labelling will be referred to as ’summarizing’.

The following research question will be answered: How to effectively use transformers multi-label text classification on legal documents that are longer than 512 tokens? To help answer the main question the following sub-questions will be answered:

- Do the truncating and summarization methods outperform the longformer method?
- Does the summarizing method outperform the truncating method?
- Which summarization method provides the best results?

It is hypothesized that the Longformer method will outperform the other methods mentioned in this research and the summarizing method will outperform the truncating method.

Chapter 2

Theoratical Foundation

This section explains the theoretical background behind the models and algorithms that are used in this research. Modelling decisions are not made in this section, these can be found in Chapter 3.

2.1 Transformers

Varshwani et al.[4] propose in their paper 'Attention Is All You Need' a model architecture that completely ignores recurrence and alternatively focuses on a self-attention mechanism to identify relationships between input and output. It is the first transduction model that relies entirely on self-attention. This mechanism allows the model to extract features for each word and determine the importance of each word relative to the other words in the sentence. Given the input variables Queries (Q), Keys (K) and Values (V), which represent linear projections, the self-attention can be computed. The dimensions of Q and K are of D_k and the dimension of V is of D_v. The softmax function is used and takes as input Q*transpose(K) divided by the square root of D_k. This is finally multiplied by V.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

Additionally, the features that are obtained from the sequences use only weighted sums and activations rather than recurrent units, making them highly parallelizable and effective. Figure 2.1 shows the encoder-decoder architecture of the Transformer model. The right side of figure 2.1 shows the encoder. This part of the

transformer model deals with mapping the sequence to a sequence input to the decoder. The decoder, which is visible on the right side of figure 2.1, receives the output of both the encoder and decoder and then generates the total output sequence.

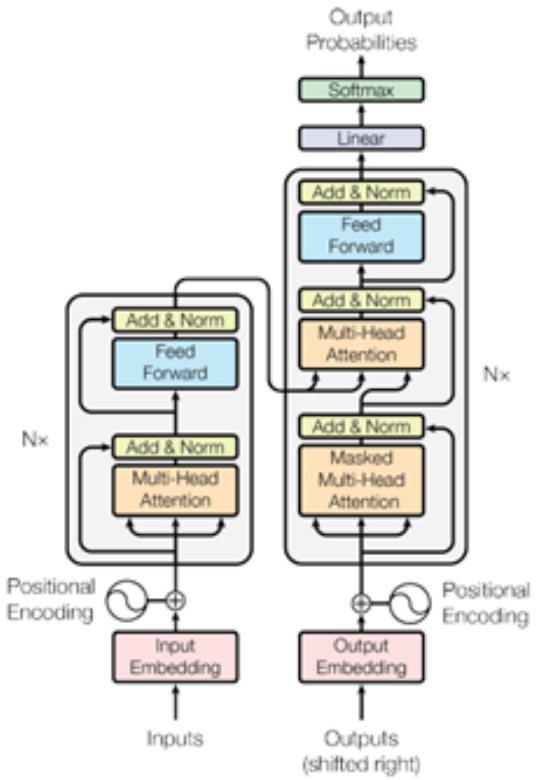


Figure 2.1: Transformer architecture [4]

2.1.1 Bert-base-cased

Devlin et al. [9] proposes a transformer model to represent language, Bidirectional Encoder Representations from Transformers (bert-base-based). Bert uses a self-attention mechanism that grows quadratically with the sequence length. Figure 2.2a shows the self-attention pattern that is followed in bert (and most other transformers). Each token in this model attends the following-up token. Every token will attend every other token if we set the length and width of this attention field equal to n (the number of tokens), resulting in O(rows$*$columns) or O(n^2) memory needs.

Bert is pre-trained on a large corpus containing English words using a self-supervised machine learning technique. This means that the model was trained without the help of humans manually labeling the data. This causes the model to be able to use tons of data that is publicly available.

What differentiates the bert model from other models is that it makes use of masked language modeling (MLM). MLM ensures that the model randomly masks 15% of the input tokens. The masked input then goes through the model again and the model has now the task to predict the masked words. This makes it possible for the model to learn the Bidirectional representation of the input sentences.

Another feature that is added to the bert model is the next sentence prediction (NSP). bert appends two masked sentences together as input during the pretraining phase. It could happen that the appended sentences are also next to each other in the original text. Bert then has to determine whether or not the two sentences followed one another. Using these two features bert is able to acquire an internal representation of the English corpus. This can then be utilized to extract features for downstream tasks such as multi-label classification.

2.1.2 Longformers

As mentioned in the introduction most transformers use a self-attention mechanism that scales quadratically with the sequence length which causes the model to not be able to process documents longer than 512 tokens. Beltagy et al[6] tackles this problem by proposing the Longformer. The Longformer model employs three complementing patterns in place of the entire attention mechanism seen in most transformer systems to support local as well as global attention:

1. Sliding window attention
2. Dilated sliding window attention
3. Global + sliding window attention

Figure 2.2b shows the sliding window pattern which is used by the Longformer model. It relies on fixed-size windows to focus on each token. With this strategy, just a select few tokens receive the full surface of attention for each token. In theory, this attention pattern appears constraint, yet it helps a multi-layer transformer creates a broad receptive field so that the top layers may develop representations that contain data from the entire input. The operation of convolutional neural networks (CNNs) and this are extremely similar [10].

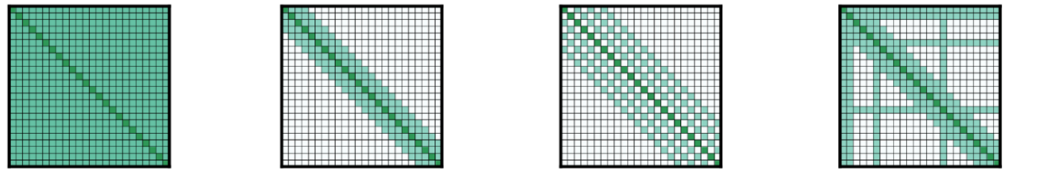


Figure 2.2: Self-attention patterns [6]

In addition to the sliding window pattern, the Longformer architecture employs a pattern to dilate the sliding surface attention as shown in figure 2.2c. In this procedure, sliding windows are expanded with gaps of variable widths. The sliding windows of a certain size are dilated with gaps of varying sizes in this method. The gaps in the kernel drive it to spread further, this enables it to capture elements in the sequence that are far from each other in a single slide without adding to the computation.

Although the previous discussed patterns are great for forming local attention they lack the versatility to establish task-specific representation. To overcome this difficulty, the Longformer architecture integrates a standard global attention mechanism. The symmetric aspect of this attention model aids in the discovery of certain sequences by taking into account all of the related tokens along the row/column in the input, thereby providing global attention to such features.

2.2 Summarization techniques

For text summarization there are two approaches one could choose: extractive and abstractive [11]. Extractive summarization is the process of creating a summary based on the most important sentence of the original text document. It does this by selecting important words and sentence and it rearranges them to create a summary. Figure 2.3 gives a schematic overview of an extractive summarizer.

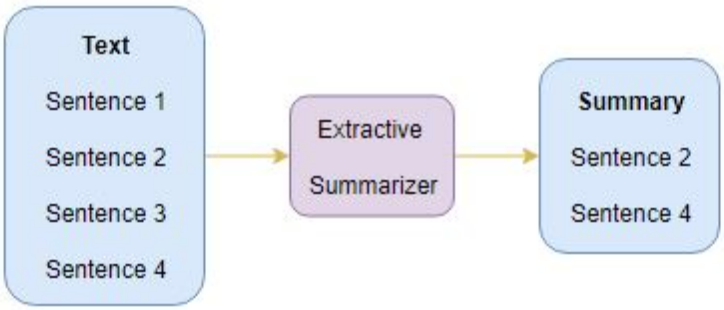


Figure 2.3: schematic overview of an extractive summarizer

Extractive summarization can be split up into both supervised and unsupervised methods. A supervised extractive summarization model would need a labelled dataset, with the labels being summaries made by humans. The supervised methods fall not within the scope of this research and thus will not be further discussed. Unsupervised methods do not require a labelled dataset to extract important features from the text. This requires sophisticated algorithms to compensate for the absence of labels.

The algorithm that is used for the unsupervised approach is called the graph-based approach. Because graphs can easily capture the structure of a document, graph-based models are widely employed in document summarization. LexRank is a graph-based technique in which the salience of the text is defined by the idea of *eigenvector centrality* [12]. This strategy is based on Google's PageRank algorithm for ranking webpages [13]. LexRank begins by constructing a graph from all of the sentences in the corpus. Every sentence represents a node, while the edges show the similarity between phrases in the corpus. In this technique, we estimate sentence similarity by treating each phrase as a bag-of-words model. This means that the frequency of word occurrence in a sentence is used to determine the similarity measure between sentences. This measurement is calculated using the TF-IDF formulation, in which phrase frequency (TF) adds to similarity strength as the number of word occurrences increases. In contrast, the inverse document frequency (IDF) considers low-frequency terms to contribute inversely to greater measurement value. This TF-IDF formulation is then used as a measure of phrase similarity.

The LexRank method calculates the relevance of sentences in a graph by comparing their importance to the importance of their neighbours, where a positive contribution raises the importance of a sentence's neighbour and a negative contribution lowers the importance of a sentence's neighbour. Figure 2.4 shows the graphical approach taken by LexRank. Si represents the sentences on the vertices and Wij are the weight on the edges. A threshold approach is used to extract the most essential phrases from the resultant similarity matrix. A threshold value is used to filter out associations between sentences whose weights are less than the threshold. The result is a subset of the similarity network from which we may select one node with the highest degree. This node is regarded as important or reflects a summary statement from the corpus

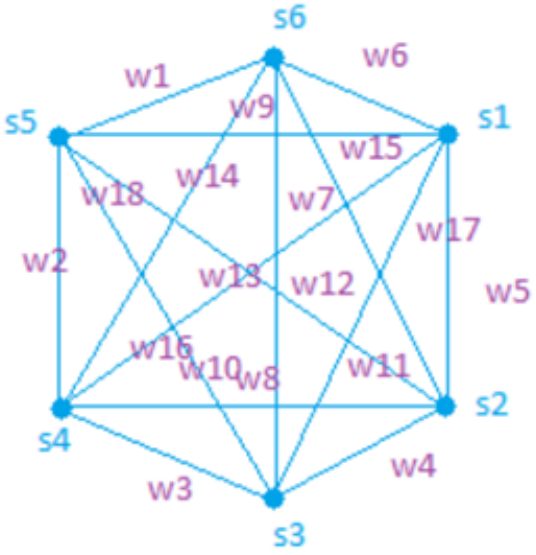


Figure 2.4: graphic approach taken by LexRank

The second approach for text summarization is abstractive. Abstractive text summarization is more similar to human summarizing. Humans read the entire text, absorb its meaning, and aim to provide the most information in the fewest phrases possible while summarizing [14]. To comprehend the original text, abstractive summarization employs linguistic approaches. The goal of this strategy is identical to that of the last one: to present information in a compact manner. It is more efficient than extractive summary because it may compress several sentences into fewer sentences by producing new sentences of its own, as a contrast to extractive summarization, which does not change the sentence structure. In comparison to extractive summarization, this involves advanced natural language processing and compression techniques. As a result, this is a more computational task when compared to the extractive summarization technique [15]. For this reason, this type of summarizing method will not be used in this research and will not be further discussed.

Chapter 3

Method

This section will give an explanation of the dataset that is used. Subsequently, a detailed overview of the steps taken to pre-process the data will be given and lastly the approaches of multi-labelling and summarizing will be explained.

3.1 The data

For this research a dataset provided by Deloitte Belastingadviseurs BV is used. This dataset is scraped entirely from Eur-Lex. Eur-Lex is an official website of European Union law and other public documents of the European Union, published in 24 official languages of the EU.

Figure 3.1 shows an sample of the Eur-Lex dataset. It contains three columns: celex, VOC CONCEPT and Text. A celex number is a document's unique identification. The language of a document does not affect the celex number. Celex numbers are assigned to the majority of papers on EUR-Lex. A celex number is made up of many elements that varies differently based on the kind of document. Every document in this dataset also contains a celex number.

The second column is named "VOC CONCEPT". This column contains the labels that are assigned to every text document. These labels were manually assigned by the Publications Office of EU [16]. The labels that are used are EUROVOC labels. EUROVOC is a multilingual thesaurus maintained by the European Union's Publications Office. The European Parliament, national and regional parliaments around Europe, several national government ministries, and other European organizations utilize it. The current edition of EUROVOC comprises almost 7,000 concepts pertaining to various EU and Member State activities (e.g., economics, health-care, trade, etc.).

The last column, Text, contains all the raw legislative documents scraped from the eur-lex website. Each document is divided into four key sections: the header, which comprises the title and name of the enforcing legal entity; the recitals, which are legal background references; and the main content, which is normally organized in articles.

	CELEX	VOC_CONCEPT	Text
0	21972A0722(03)	Switzerland|trading operation|customs duties|t...	21972A0722(03) Agreement between the European ...
1	21980D1231(03)	Greece|tariff policy|agreement (EU)|accession ...	21980D1231(03) Decision No 3/80 of the EEC-Ice...
2	21981A0710(02)	Hungary|trade agreement (EU)|goatmeat|sheepmea...	21981A0710(02) Exchange of letters between the...
3	21986A1115(03)	trade agreement|Portugal|protocol to an agreem...	15.11.1986 EN Official Journal of the European...
4	21987A0720(02)	protocol to an agreement|customs harmonisation...	20.7.1987 EN Official Journal of the European ...

Figure 3.1: Sample of the Eur-lex dataset

Table 3.1 gives an overview of the most important features of the original and limited dataset. The original dataset contains 257,816 entries and the total amount of labels is 7,033. The dataset contains a large number of entries and therefore also an extreme amount of labels. It was decided to reduce the extreme amount of labels to only the 91 most used labels. This decreases data sparsity and achieves higher accuracy. Figure 3.2 shows a bar plot display of the number of labels and their frequency. For instance, 1,313 of the labels are used between 1 and 10 times and 400 of the labels are used between 1,000 and 2,500 times. In the limited dataset, only the labels that appeared between 5,000-10,000 and 10,000-15,000 were used. This makes up for a total amount of 91 labels. The average document length is 2,567 tokens. This is considered a long document for most transformers like bert.

Features	Amount original dataset	Amount limited dataset
Amount of entries	257,816	257,816
Amount of labels	7,033	91
Average document length	2,567	2,567

Table 3.1: Features of the original and limited dataset

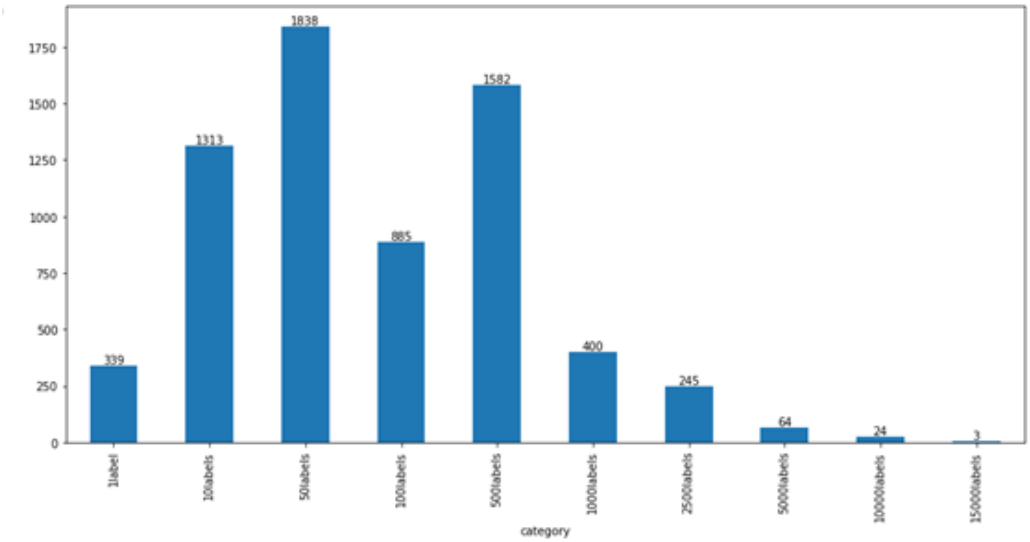


Figure 3.2: Bar plot of label frequency

3.1.1 Pre-processing the data

Preprocessing is an essential step to ensure that the algorithm can handle the data correctly. It cleans the data and prepares it for analysis. The following steps were taken to prepare the text for multi-labelling:

1. Removal of html tags and other special characters such as punctuation marks and non-alphabetic characters (removal of special characters is not used in the pre-processing of the data for the summarization method).
2. Remove stop words from the text, this is done using the default stop word list from the NLTK Library [17].
3. Stemming, this means reducing every word to their stem. For example riding and ride would become rid.

After pre-processing the text the dataset was exported and it is going to be used in the methods that will be described in the next sections.

3.2 Method 1 - Truncating

The first method that was used to multi-label the documents was to only use the first 512 tokens of each document. This technique was chosen because this is the most basic form of using Transformers. Most transformers automatically truncate all the documents that are longer than 512 token to a length of 512.

To train and test the models, the dataset is divided into three independent datasets as shown in table 3.2. The training set is used to train and discover any hidden patterns and features in the data. The same training data is supplied to the model repeatedly in each epoch, and the model continues to learn the data's features. The training set contains 122,401 data entries, which is the largest portion of the dataset. The validation set contains 30,601 entries. This set is used to validate the performance of the transformer during training. The test set contains 38,251 entries, this set gives the final performance metrics.

Train	Validation	Test
122,401	30,601	38,251

Table 3.2: Train, validation and test sets

The bert transformer was used to multi-label the data. This model was chosen because of multiple reasons. Bert has been pre-trained on enormous amounts of data. This is an advantage because it increases accuracy[18]. Another reason is that bert takes the context of the words of the sentences into account. This means that bert returns different vectors for the same words based on the context. For example, bert would return different word embeddings for the words 'bat' in the sentences "The bat flew in the sky" and 'He hit the ball with a bat'. This, is because it looks at the words around the word "bat", whilst other methods, like RNN's would return the exact same word embedding for the word "bat".

The parameters that were chosen for training are 12 for epochs and 8 for batch size. The epochs were chosen based on trial and error. The general rule of thumb is to start with an epoch value of three times the amount of columns in your dataset. Which in this case is 3*3=9. After trying a few different values it was determined that 12 resulted in the highest accuracy and the model was then trained using these parameters.

3.3 Method 2 - Summarizing

The next method that was chosen to use is to summarize the long documents to 512 tokens in order for the bert transformer to process it. It was important to find an unsupervised model for this task. This is because the dataset used in this research does not contain any labelled summaries. Moratanch et al. [19] researched different extractive summarization methods, a summary of this can be found in Appendix A. The unsupervised graph-based approach has the advantage that it captures redundant information and it improves the coherency, meaning the overall understandability of the text.

Table 3.3.1 shows an overview of all the summarizers that were used. A comparison between the results of the summarizers was made by looking at the difference in the resulting summaries. Table 3.3.2 illustrates a comparison of the output results by the summarizers. For example, 42 out of the 100 sampled documents are the exact same in the outputs of Lex Rank and Luhn. Noticeable is that almost all the summarize output are the same summaries. This means that there is no noticeable difference between the summarizers. The Gensim summarizer was chosen because this summarizer had the best time performance.

Summarizing method	Description
Gensim (unsupervised)	Gensim summarizes based on the ranks of the sentences within a text. It used a variation of the TextRank algorithm. Text rank is a graph-based ranking algorithm.
Lex Rank (unsupervised)	A sentence that is similar to many other sentences in the text is likely to be significant. LexRank takes the notion that a given sentence is suggested by other comparable sentences and hence ranks higher.
Latent Semantic Analysis (LSA) (unsupervised)	an unsupervised learning method suitable for extractive text summarizing It finds semantically meaningful phrases by applying singular value decomposition (SVD) to a term-document frequency matrix.
Luhn	The TF-IDF technique underpins the Luhn Summarization algorithm (Term Frequency-Inverse Document Frequency). It is effective when both extremely low frequency words and highly frequent words (stop words) are insignificant
Kl-sum	It chooses sentences based on their word distribution resembling to the original text. It seeks to reduce the KL-divergence criterion. It employs a greedy optimization method and continues to add phrases until the KL-divergence reduces.

Table 3.3.1: Extractive summarization methods

Summary method	Summary method	Amount of duplicates
Kl-sum	LSA	37
LexRank	Luhn	42
LSA	KL-sum	37
Gensim	LexRank	34
Gensim	LSA	35
Luhn	Gensim	34
Kl-sum	Luhn	37
LSA	Luhn	41
LexRank	LSA	42
Luhn	LSA	41
Luhn	KL-sum	37
LexRank	Kl-sum	37
LexRank	Gensim	34
Gensim	Kl-Sum	34
Kl-Sum	LexRank	37
LSA	Gensim	35

Table 3.3.2: Comparison of the extractive summarizing methods

The summarization with Gensim is performed on all documents in the eur-lex dataset. Figure 3.3 shows the length ration between the original and summarized documents. Around 40.000 documents have a ratio of 1.0. This means that there was no change in the document length after summarizing. The reason for this is because a portion of the documents was already 512 tokens or less and the Gensim model was set to summarize every document to 512 tokens. This results in no changes within the length of those documents.

The summarization of all the documents along with the labels was then given as input to the bert transformer. To get an accurate comparison of the summarizing method and the truncating method the same parameters for epoch and batch were chosen for the summarizing method as described in section 3.2.

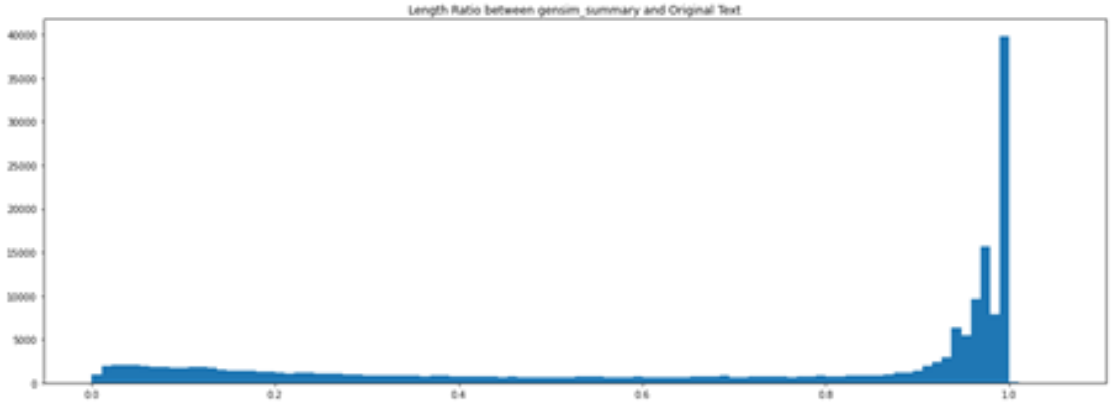


Figure 3.3: Ratio of length between original and summarized documents

3.4 Method 3 - Longformers

The final method that was chosen to compare the other multi-label methods with is Longformer. This type of transformer was chosen because, as mentioned before, it can process up to 4,096 tokens. There are other transformers that can process 4096 tokens like Big Bird. Although Big Bird outperforms Longformer by 3-4 percent, it takes up to 16 times computing power [20].

The first step was to prepare the data labels for the Longformer. This is done by using one hot encoding (ohe). This is the process of turning categorical labels into binary integers. Where the value of 1 of a feature corresponds to the original label. The dataset was then split into a training and test set. The dataset was then split into a train, validation and test set. The same amount of entries for each of the sets was chosen as described in section 2.2. The Longformer was then trained using 12 epochs and a batch size of 8.

Chapter 4

Results

This section will give an overview of the results of the three used methods. The first section gives an overall summary and performance of the three used methods.

4.1 Overall performance and results

The output of all models was exported as a .csv file. Appendix A contains samples of all the export files. The export file contains 7 columns with the first column being the CELEX number that is unique to the document. The next five columns contain the top 5 predicted labels by bert. Each of these columns contains a tuple with the first index being the predicted label and the second index is the chance of the predicted label being correct. The last column contains the original label(s).

The precision, recall and F1 score were used to evaluate the performance of the methods. These metrics are calculated using true positive, false negatives and false positives. The following formulas are used to calculate the metrics:

1. $\text{Accuracy} = \dfrac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$
2. $\text{Precision} = \dfrac{\text{TP}}{\text{TP} + \text{FP}}$
3. $\text{Recall} = \dfrac{\text{TP}}{\text{TP} + \text{FN}}$
4. $\text{F1-score} = \dfrac{\text{2 x Precision x Recall}}{\text{Precision} + \text{Recall}}$

To determine whether the label belongs to a class a threshold is used. Every label that has a probability that is larger than the threshold is mapped to a class

and everything that falls under the threshold is mapped to the other class. In this research, the classes are defined as 0 and 1 meaning a label is or is not assigned to a document. The default value is set to 0.5 but because this dataset is highly imbalanced, an optimal threshold must be determined. Figure 4.1 displays the ROC curve lines for all the multi-label methods. The black dot is placed on the curve in the line where the optimal threshold is.

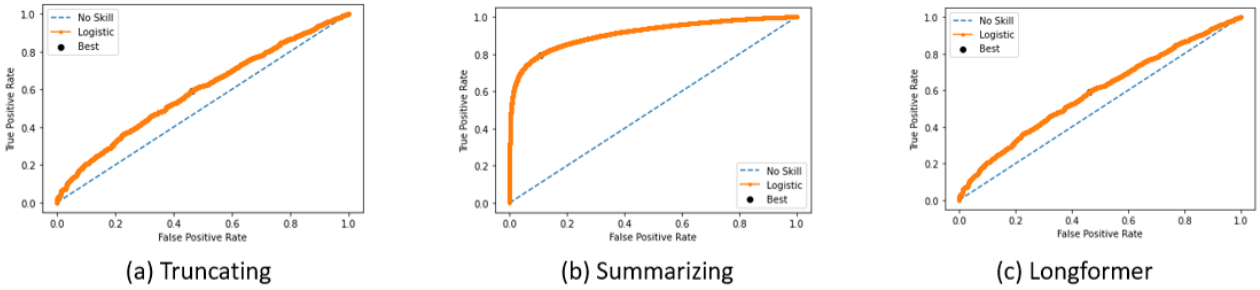
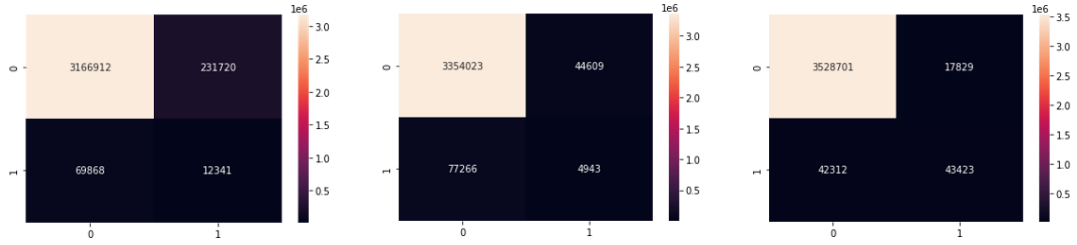


Figure 4.1: ROC curves to determine the optimal threshold values

Figure 4.2 displays the confusion matrices of the multi-labelling methods. The matrices give a summary of the predictions that were made by the model. The top left corner of the model represents the true negatives, which is when the model correctly predicts when a label is not true for a document. In the top right corner are the false positives. This is when the model incorrectly assigns a label to a document. The bottom left corner are the false negatives which is when the model predicts that a label is not there when it actually is. In the bottom right corner are the true positives, this is when the model correctly identifies a label.

The multi-labelling method that identified the most amount of true positives is the Longformer. It correctly identified 43.423 labels which is almost 10 times as high as the summarizing method and 3 times as high compared to the truncating method.



(a) Truncating (b) Summarizing (c) Longformer

Figure 4.2: Confusion matrices of the three multi-labelling methods

Table 4.3 shows the recall, precision and F1 score for the three multi-labelling methods. These metrics provide a lot of information because they give knowledge about both the quality and quantity of our data and model. The precision is used to calculate the frequency the model was able to correctly predict a positive class. For instance, the precision for the truncating method is 0.26, this means that when the model predicts a label it is correct 26 out of 100 times. The recall represents how many labels the model was able to identify out of all possible labels. For example, the recall is 0.6 for the truncating method. This means that the model was able to correctly predict 6 out of 100 labels. The last metric that is used is the F1 score which is the harmonic mean of the precision and recall.

Method	Precision	Recall	F1 Score
Truncating	0.26	0.06	0.09
Summarizing	0.05	0.15	0.08
Longformer	0.71	0.51	0.59

Table 4.3: Metric summary of the three multi-labeling methods

Chapter 5

Discusion

In this chapter, the results of the three methods are evaluated and compared.

5.1 Result evaluation

The models will be evaluated on the F1 score. The reason for this is that the dataset is highly imbalanced. The total amount of support that is used within class 0 is 3,398,632 and for class 1 it is 82,209 as shown in table 5.1. The support represents the number of times the label was or was not predicted. For example, for any document in the dataset there were 91 possible labels that could be assigned to the document. The vector of that document could look like [0,0,1,0,1....0,1]. The 0 and 1 represent actual labels, the label is 1 if the model predicts that the label belongs to the document and 0 otherwise.

Class	Support
0	3,398,632
1	82,209

Table 5.1: Support of the classes

The F1 score is a popular evaluation metric for machine and deep learning models when dealing with imbalanced datasets [21]. It combines the precision and recall score. It is also called the harmonic mean of precision and recall [22]. Table 5.2 shows an overview of the F1 scores of the different multi-labelling methods.

Method	F1 Score
Truncating	0.09
Summarizing	0.08
Longformer	0.59

Table 5.2: F1 scores of the three multi-labelling methods

The truncating method and the summarizing methods have almost equal F1 scores. Initially, it was expected that the summarizing method would outperform the truncating method. This is because it was expected that the summary would be more information dense.

The reason for the low score of the summarizing method could be that the Gensim summarizer did not perform well since most of the documents are too unstructured for the model to process well. For example, a large number of documents do not contain any interpunction. This means that the summarizer will interpret this as one big sentence. Since the summarizer looks at the document, sentence-wise, this will result in bad outputs.

The truncating method slightly outperformed the summarizing method by 0.01. Although this is not a significant amount it performed much better than initially expected. The reason for this could be that a large number of documents begin with a summary of the legal document. It could be that some of the labels were given to this document based on the short summary at the beginning of the document.

A possible factor that could have influenced the F1 scores of the truncating and summarizing methods, is that the bert model was able to predict labels that were accurate for the document but might have been missing in the original label set. The documentalists who labelled the original documents on eur-lex might have missed a label that bert was able to identify correctly. A possible way to prove this is to ask a board of experts to look at the labels provided by the bert model and determine whether or not bert was correct.

The longformer has an F1 score of almost 7 times as high as the other two methods. With a precision of 0.71, it was able to predict a label correctly 71 out of 100 times. Although this is a high score compared to the other methods, relatively it is not an outstanding score. Figure 5.1 shows the general rule of thumb for an F1 score to be considered good. The reason for the average F1 score for the longformer method could be that the parameters of the models, epoch and batch,

were not optimized enough. Due to time constraints, it was not possible to train the model multiple times with different parameters.

F1 score	Interpretation
> 0.9	Very good
0.8 - 0.9	Good
0.5 - 0.8	OK
< 0.5	Not good

Figure 5.1: Interpretation of F1 scores

Table 5.3 shows the training times for all the used methods. Due to the high training times, it was not possible to try out different hyper-parameters such as epoch and batch for each model. Fine-tuning the parameters could increase the performance of all the models.

Method	Training time (hours)
Bert-Base-Cased on truncated documents	24
Bert-Base-Cased on summarized documents	24
Longformer	72
Gensim text summarization	26

Table 5.3: Train, validation and test sets

Chapter 6

Conclusion

This research aimed to find a method to multi-label long documents within the legal domain with transformers. The issue that it tried to tackle was the limited amount of 512 tokens a transformer can process. A new method was introduced to address this issue – summarizing the documents to 512 tokens before multi-labelling it with the bert transformer. This method was then compared with two already existing methods to multi-label documents with transformers: truncating the data after 512 tokens and Longformers.

The decision to use the bert model was made because it was trained on a lot of data and it takes the context of the sentences into account. Gensim algorithm was chosen to summarize the data because this proved to be the most optimal model for this specific dataset. And lastly, the Longformer model was chosen because it can process up to 4096 tokens and has the least amount of computing power when compared to Big Bird.

After performing multi-labeling with the three methods; truncating, summarizing and Longformer the research question and the subquestions can be answered. With the first sub-question, we tried to determine whether the bert transformer method would outperform the longformer method. The performance is measured in the F1 score and it is clear that the first two methods, truncating and summarizing which used the bert transformer, did not outperform the Longformer. The Longformer achieved an F1 score of 0.59 which is almost 7 times as good as the other two methods.

The second sub-question is whether the summarizing method would outperform the truncating method. Although it was hypothesised that the summarizing method would be better there was no clear difference in performance for both

methods. The truncating method achieved an F1 score of 0.09 and the truncating method a score of 0.08. The reason the truncating method performed better than initially thought could be that a large portion of the documents begins with a short summary of the document. The assigned labels could be based on these summaries causing the transformer to perform better than what was expected.

The last subquestion tried to find the best summarization method to summarize the documents to 512 tokens. It was determined that an unsupervised graphic approach would be the best method for this dataset because the dataset did not contain any labels for summaries. Gensim was chosen because this algorithm had the least amount of computing time in comparison to the other summarization technique that was used to compare.

With the answers to the sub-question, the main question can be answered. Looking at the performance of the researched methods it is clear that the Longformer provides the best results when multi-labelling long documents. Although the summarization method did not perform well on the eur-lex dataset it could be promising for datasets with more structured documents which was not the case in the eur-lex dataset.

Chapter 7

Future work

There are several possible improvements that can be made to this research. Although bert transformer is pre-trained on a lot of data it could be better to use a transformer that is pre-trained on solely data within the legal domain to increase performance. There currently is no type of Longformer that is pre-trained on legal data, but there is a transformer that limits the token amount to 512 that is pre-trained on legal data [23]. Another improvement could be to summarize the data using an abstractive method such as Lonformer. Although abstractive summarization is a computationally expensive task it could improve performance because it is pretrained and it is able to capture patterns between longer sequences. The capturing of longer sequences to sequence mappings could be ideal for this specific dataset because it tackles the issue of the documents containing long sentences due to little use of interpunction.

Another interesting approach to increase performance on the summarizing method is to use longformer to summarize the documents. This is because the longformer is pretrained which could increase the accuracy of the summaries. Another interesting approach to multi-label the long documents is to divide the documents into chunks of any size and analyze if different chunk sizes influence the output. The reason this could give interesting results is that important information could lose within the process when chunking the text. For example, if you choose to use a chunk size of 400 then it could be the case that there contextual information was between the tokens 300 and 500. But this would have been lost because the chunk size was set to 0-400. Analyzing different chunk sizes could therefore be an interesting approach.

Additionally, chunk-wise summarization could be interesting to research. Figure 7.1 illustrates the process of this type of summarization on a document of 5100

tokens. To summarize a document of 5100 tokens, it could be divided up into chunks of 510 tokens. This could then be summarized with a transformer-based model to 51 tokens. The next step is to append all the summarizations of 51 tokens which will result in a summarized document of 510 tokens. A transformer can then be used to multi-label the summarized documents. This idea could also be generalised to summarizing with a longformer. This would give the opportunity to summarize and multi-label longer documents.

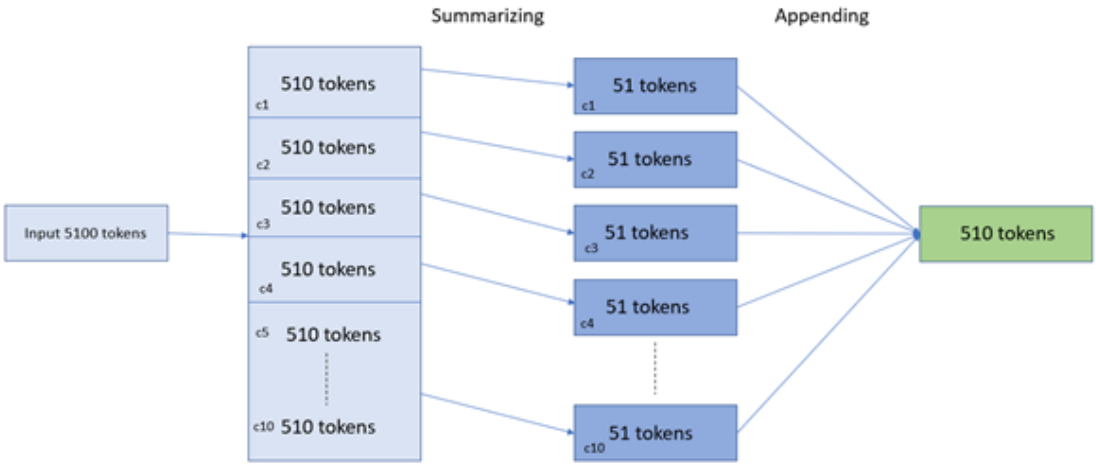


Figure 7.1: Chunk-wise summarization

Bibliography

[1] Roshan Kumari and Saurabh Kr Srivastava. "Machine learning: A review on binary classification". In: *International Journal of Computer Applications* 160.7 (2017).

[2] Mohamed Aly. "Survey on multiclass classification methods". In: *Neural Netw* 19.1 (2005), p. 9.

[3] Grigorios Tsoumakas and Ioannis Katakis. "Multi-label classification: An overview". In: *International Journal of Data Warehousing and Mining (IJDWM)* 3.3 (2007), pp. 1–13.

[4] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[5] Qizhe Xie et al. "Unsupervised data augmentation for consistency training". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 6256–6268.

[6] Iz Beltagy, Matthew E Peters, and Arman Cohan. "Longformer: The long-document transformer". In: *arXiv preprint arXiv:2004.05150* (2020).

[7] Manzil Zaheer et al. "Big bird: Transformers for longer sequences". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 17283–17297.

[8] Mandar Joshi et al. "Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension". In: *arXiv preprint arXiv:1705.03551* (2017).

[9] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[10] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. "Understanding of a convolutional neural network". In: *2017 international conference on engineering and technology (ICET)*. Ieee. 2017, pp. 1–6.

[11] Udo Hahn and Inderjeet Mani. "The challenges of automatic summarization". In: *Computer* 33.11 (2000), pp. 29–36.

[12] Günes Erkan and Dragomir R Radev. "Lexrank: Graph-based lexical centrality as salience in text summarization". In: *Journal of artificial intelligence research* 22 (2004), pp. 457–479.

[13] Lawrence Page et al. *The PageRank citation ranking: Bringing order to the web.* Tech. rep. Stanford InfoLab, 1999.

[14] Som Gupta and Sanjai Kumar Gupta. "Abstractive summarization: An overview of the state of the art". In: *Expert Systems with Applications* 121 (2019), pp. 49–65.

[15] Vishal Gupta and Gurpreet Singh Lehal. "A survey of text summarization extractive techniques". In: *Journal of emerging technologies in web intelligence* 2.3 (2010), pp. 258–268.

[16] *MS Windows NT Eu Publication Office*. `https://publications.europa.eu/en`. Accessed: 2022-06-30.

[17] *NLTK NLTK Libaray*. `https://www.nltk.org/`. Accessed: 2022-05-20.

[18] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. "Using pre-training can improve model robustness and uncertainty". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 2712–2721.

[19] N Moratanch and S Chitrakala. "A survey on extractive text summarization". In: *2017 international conference on computer, communication and signal processing (ICCCSP)*. IEEE. 2017, pp. 1–6.

[20] Chuhan Wu et al. "Hi-Transformer: hierarchical interactive transformer for efficient and effective long document modeling". In: *arXiv preprint arXiv:2106.01040* (2021).

[21] László A Jeni, Jeffrey F Cohn, and Fernando De La Torre. "Facing imbalanced data–recommendations for the use of performance metrics". In: *2013 Humaine association conference on affective computing and intelligent interaction*. IEEE. 2013, pp. 245–251.

[22] Juri Opitz and Sebastian Burst. "Macro f1 and macro f1". In: *arXiv preprint arXiv:1911.03347* (2019).

[23] Ilias Chalkidis et al. "LEGAL-BERT: The muppets straight out of law school". In: *arXiv preprint arXiv:2010.02559* (2020).

Appendices

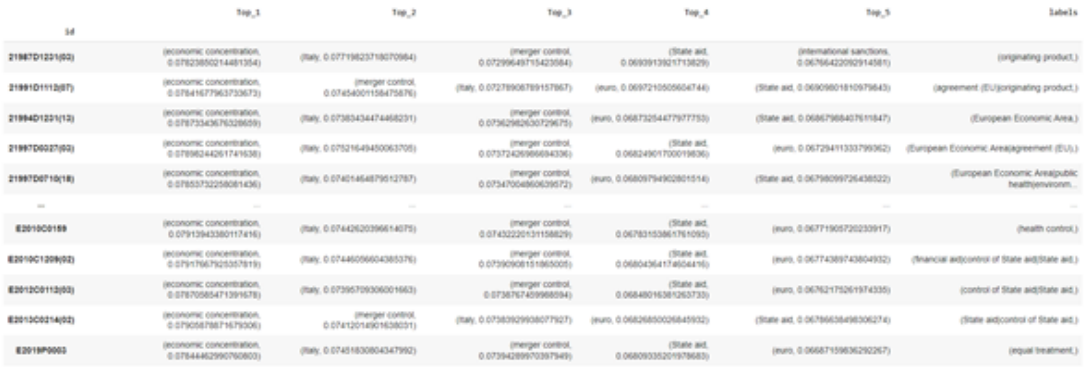
Appendix A

Categories	Methodology	Concept	Advantages
SUPERVISED LEARNING APPROACHES	Machine Learning approach Bayes rule	Summarization task modelled as classification problem	Large set of training data improves the sentence selection for summary
SUPERVISED LEARNING APPROACHES	Artificial Neural Network	Trainable summarization - neural network is trained , pruned and generalized to filter sentences and classify them as "summary" or "non-summary sentence"	The network can be trained according to the style of human reader. The set of features can be altered to reflect user's need and requirements
SUPERVISED LEARNING APPROACHES	Conditional Random Fields (CRF)	Statistical modelling approach which uses CRF as a sequence labelling problem	Identifies correct features and provides better representation of sentences and groups terms appropriately into its segments
UNSUPERVISED LEARNING APPROACHES	Graph based Approach	Construction of graph to capture relationship between sentences	1) Captures redundant information 2)Improves coherency
UNSUPERVISED LEARNING APPROACHES	Concept oriented approach	Importance of sentences calculated based on the concepts retrieved from external knowledge base(wikipedia, HowNet)	incorporation of similarity measures to reduce redundancy
UNSUPERVISED LEARNING APPROACHES	Fuzzy Logic based approach	Summarization based on fuzzy rule using various sets of features	improved quality in summary by maintaining coherency

Figure 2: Different summarization techniques [19]

id	Top_1	Top_2	Top_3	Top_4	Top_5	Labels
21987D1231(02)	(economic concentration, 0.07606174051761627)	(merger control, 0.07524641603231 43)	(Italy, 0.07129500806331635)	(State aid, 0.06973601877689362)	(euro, 0.06711452454328537)	(originating product,)
21991D1112(07)	(economic concentration, 0.07572752231842994 7)	(merger control, 0.07539654523134232)	(Italy, 0.07287457865334778)	(State aid, 0.06978736072778702)	(originating product, 0.06686054915189743)	(agreement (EU)originating product,)
21994D1231(12)	(economic concentration, 0.07606831388509064)	(merger control, 0.07543617486955735)	(Italy, 0.07328099757432938)	(State aid, 0.07001964747905731)	(originating product, 0.06669083237648 01)	(European Economic Area,)
21997D0327(03)	(economic concentration, 0.07608957588672638)	(merger control, 0.07575301080942154)	(Italy, 0.07385237514972667)	(State aid, 0.07029347121715546)	(originating product, 0.06683906912803 65)	(European Economic Area)agreement (EU),)
21997D0710(18)	(economic concentration, 0.07526247406005 86)	(merger control, 0.07563973218202991)	(Italy, 0.07196379188776016)	(State aid, 0.06969413161277771)	(originating product, 0.06627502292394638)	(European Economic Area)public health)environm...
...	...	...	...	...	...	...
E2010C0189	(economic concentration, 0.07630062848329544)	(merger control, 0.07418380677700043)	(Italy, 0.07362616035892968)	(State aid, 0.07064836472272873)	(originating product, 0.06755146384239197)	(health control,)
E2010C1209(02)	(economic concentration, 0.07620751557757568)	(merger control, 0.07534019649028778)	(Italy, 0.07374043762683868)	(State aid, 0.07018783686545227)	(originating product, 0.06795982271432877)	(financial aid)control of State aid)State aid,)
E2012C0112(03)	(economic concentration, 0.07614116370677948)	(merger control, 0.07479923963546753)	(Italy, 0.07311459630727768)	(State aid, 0.07040536403656006)	(originating product, 0.06779059022665024)	(control of State aid)State aid,)
E2013C0214(02)	(economic concentration, 0.07607914507369069)	(merger control, 0.07467909992334213)	(Italy, 0.07184892892837524)	(State aid, 0.06967880915403366)	(originating product, 0.06711931529660919)	(State aid)control of State aid,)
E2019P0003	(economic concentration, 0.07723207026720047)	(merger control, 0.07590036029418869)	(Italy, 0.07091659007479858)	(State aid, 0.06929625570774076)	(euro, 0.06686057150363922)	(equal treatment,)

Figure 3: Sample of the truncating method export file



id	Top_1	Top_2	Top_3	Top_4	Top_5	Labels
21987D1231(02)	(economic concentration, 0.07823850214481354)	(Italy, 0.07719623718070964)	(merger control, 0.07299649715423584)	(State aid, 0.06939139217113629)	(international sanctions, 0.06766422092914581)	(originating product,)
21991D1112(07)	(economic concentration, 0.07841677963733673)	(merger control, 0.07454001158475876)	(Italy, 0.07278908789157867)	(euro, 0.06972105056047 44)	(State aid, 0.06909801810979643)	(agreement (EU)originating product,)
21994D1231(12)	(economic concentration, 0.07873343676328659)	(Italy, 0.07383434474468231)	(merger control, 0.07362982630729675)	(euro, 0.06873254477977753)	(State aid, 0.06879684076 11847)	(European Economic Area,)
21997D0327(03)	(economic concentration, 0.07898624261741638)	(Italy, 0.07521649450063705)	(merger control, 0.07372426966694336)	(State aid, 0.06824901700019836)	(euro, 0.06729411333799362)	(European Economic Area)agreement (EU),)
21997D0710(18)	(economic concentration, 0.07855732258081436)	(Italy, 0.07401464879512787)	(merger control, 0.07347004860639572)	(euro, 0.06809794902801514)	(State aid, 0.06798099726438522)	(European Economic Area)public health)environm...
...	...	...	...	...	...	...
E2010C0189	(economic concentration, 0.07913943380117416)	(Italy, 0.07442620396614075)	(merger control, 0.07432220131158829)	(State aid, 0.06783153861761093)	(euro, 0.06771905720233917)	(health control,)
E2010C1209(02)	(economic concentration, 0.07917667925357819)	(Italy, 0.07446056604385376)	(merger control, 0.07390908151865005)	(State aid, 0.06804364174604416)	(euro, 0.06774389743804932)	(financial aid)control of State aid)State aid,)
E2012C0112(03)	(economic concentration, 0.07870565471391578)	(Italy, 0.07395709306001663)	(merger control, 0.07387674599988594)	(State aid, 0.06848016381263733)	(euro, 0.06762175261974335)	(control of State aid)State aid,)
E2013C0214(02)	(economic concentration, 0.07905878671679306)	(merger control, 0.07412014901638031)	(Italy, 0.07383929938077927)	(euro, 0.06826850026845932)	(State aid, 0.06786638498306274)	(State aid)control of State aid,)
E2019P0003	(economic concentration, 0.07844462990760803)	(Italy, 0.07451830804347992)	(merger control, 0.07394289970397949)	(State aid, 0.06809335201978683)	(euro, 0.06687159836292267)	(equal treatment,)

Figure 4: Sample of the summarizing method export file

id	Top_1	Top_2	Top_3	Top_4	Top_5	Labels
32018R2382	(Germany, 0.23919368665184784)	(United Kingdom, 0.05186885595321655)	(Belgium, 0.03413353487849235 5)	(Portugal, 0.03222393244504928 6)	(catch quota, 0.02961139119466733 93)	(Germany,)
31974D0033	(provision of services, 0.06861675326824188)	(originating product, 0.05381679966744423)	(Belgium, 0.04862241446971893)	(European Commission, 0.04690804708552360 5)	(action for annulment of an EC decision, 0.045...	(single market,)
62020MB814(01)	(Belgium, 0.76126694679260 25)	(originating product, 0.71817899675030518)	(anti-dumping duty, 0.14203326404094696)	(Portugal, 0.07784065460945 72)	(EU Member State, 0.04502130299806095)	(originating product, Belgium)
32005D0748	(import (EU), 0.14908596873283386)	(pip fruit, 0.12663353975585555)	(tariff quota, 0.08575364202260971)	(EU aid, 0.07422150671482086)	(European trademark, 0.07409352809190 75)	(pip fruit, technical standard)
22006D0078	(EU programme, 0.22983604345130 92)	(human rights, 0.11834235489368439)	(Germany, 0.08792395889759064)	(market approval, 0.08257915824651718)	(import licence, 0.06478159129619596)	(fishing area,)
31989D0177	(provision of services, 0.29862982035468323)	(action for annulment of an EC decision, 0.196...	(European trademark, 0.08343714475631714)	(equal treatment, 0.07185305419826415)	(European Commission, 0.05134843289852142)	(export refund, provision of services)
31987R2930	(provision of services, 0.04006188735365876)	(European trademark, 0.03646623715758324)	(tariff quota, 0.03501481935361889)	(EU aid, 0.03005935810506344)	(European Commission, 0.02905153855681419 4)	(CCT duties,)
92001E002489	(EU aid, 0.45200684342193604)	(health control, 0.08956689367559891)	(economic sanctions, 0.07243645191192627)	(registered trademark, 0.05672165378928184 5)	(equal treatment, 0.05092520266771316 5)	(EU aid,)
62006TB0058	(EU aid, 0.5486142039299011)	(provision of services, 0.25419256099111786)	(action for annulment of an EC decision, 0.191...	(health control, 0.17833292484283447)	(European trademark, 0.12689215474491 12)	(provision of services, technical standard, EU...
62015TA0061	(registered trademark, 0.71369522809956 23)	(European Economic Area, 0.49018314480781555)	(protocol to an agreement, 0.39932221174240 11)	(aid to agriculture, 0.14282950758934 02)	(health control, 0.10362966356661652)	(protocol to an agreement, registered trademar...

Figure 5: Sample of the Longformer method export file