DEVELOPMENT OF DOCUMENT CLASSIFICATION SYSTEM USING LAYOUTLMv2

Abstract

In this work, Document classification is selected as the target application for LayoutLMv2 model that consists of both text and layout knowledge. First, the target sources of text are selected in a way which allows for obtaining the most diverse documents that would represent as many types of documents as possible. Particular attention is paid to research papers where delivering components such as tables and figures are often excluded during preprocessing phase. The files shall first be rationed by picking out the related text and layout data which would prepare them for model training.

After data preprocessing the LayoutLMv2 model is trained to improve the model's capacity to classify documents according to its context and structure. Layout data together with textual information give the model an additional possibility – to understand the internal dependencies of documents and thus be more precise in classification. To evaluate the result from the model maintenance, key measures such as the rate of accuracy and the positive margins of precision are employed to guarantee that the model returns consistent and sound proposition.

This approach goes beyond the conventional document classification by using LayoutLMv2 with its exceptional features. Due to the use of the combination of text and layout information the given research helps in creating better and more precise systems in document understanding. It is evident from the results that the integration of these two data sources can provide insight for the future developments in document classification.

Table of Content

List of Figures

Chapter 1

Introduction

Conventional techniques still hold good for document attributes that are highly ordered; but on the other hand, the exists a sea of information that is literally unmanageable. This makes it contextually appropriate to apply advanced procedures that can identify structures in a document layout and the documents' content itself. Technological breakthroughs in recent years connected to AI have included new features that present deeply philosophical and accurate text analysis for NLP. Nevertheless, layout-sensitive transformers such as LayoutLMv2 have been developed that perform highly in extract-text and text recognition based on the document. Considering the concepts from the Layout-LM model, LayoutLMv2 focuses on the integration of visual and text information for better understanding of the document presentation and content to increase classification performance (Xu et al., 2020). Layout-aware transformers dovetail with document classification, which enhances their accuracy and time because transformers can manage large quantities of data without involving people. This proposal aims at evaluating LayoutLMv2's performance regarding document classification, assuming that this model is more accurate than other existing models. While the existing literature tends to present models' theoretical findings, this proposal seeks to shed light on practical steps and results of employing LayoutLMv2, hence advancing the advancement of better AI-based info management systems. This research will present a method for document classification by means of artificial intelligence algorithms.

1.1 Project Background

Document classification has always been a part of information management because someone has had to sort and store, retrieve, and process document collections from an early age. Most of the initial methods employable for WSD are rule-based schemes or fundamental learning models and worked effectively only for the textual information part of the documents, as they performed dismally when examined against the layouts of more intricate documents or the documents containing both content and images or other forms of materials (Sebastiani 2002). Such restrictions are most noticeable in cases where the structure itself contains highly semiotic features, including invoices, contracts, and academic papers. The new developments in the deep learning technique – in conjunction with the transformer model- have greatly enhanced the application of Natural Language Processing (NLP). The transformer model described in the paper by Vaswani et al. (2023) was widely used in the subsequent scientific work devoted to understanding and generation of human language and had become a steppingstone to the next level of improvement in the field. Expanding on this, BERT (Devlin et al., 2019) and GPT3 (Brown et al., 2020) type models are very popular for most NLP tasks, including language translation and text summarization. However, these models mainly focus on the text body of the documents and rarely pay attention to the arrangement of documents.

To overcome these drawbacks, something better has been designed known as LayoutLMv2 which is a layout aware transformer that provides the content of the text and the images together into consideration (Xu et al. , 2020). This model considers text, layout, and image as input characteristics which enrich the structural characteristics analysis of documents. This kind of approach is useful in contexts where the position of the information is important for comprehension of the textual material. For example, in forms and tables the layout offers useful hints that are critical for correct classification of subtypes. This project is justified by the current inherent issues within document classification methods and the use of layout-aware transformers is proposed as a solution. Therefore, this research will endeavor to concentrate on LayoutLMv2 as an output of text-only models to meet the signals' increased demands in the document classification field. The hypothesis that set up this research is that layout information will add value to the model hence to the efficiency of the information management in organization.

The first document classification systems primarily used rule-based approaches, meaning that numerous inputs of human effort were needed to develop rules for the classification and this methodology had fixed weaknesses through lack of flexibility when it comes to the emergence of new types of documents or alteration of the classification paradigms. While with the help of machine learning models were able to learn and enhance their performance through the course of time, they were mostly oriented on textual data. This focus on text alone became inadequate for documents where layout and structure of the document are part of the comprehension of text. For instance, tables in invoices may have their structure in which a position of a number has a certain meaning, contracts that provide information about sections or clauses' legal implications. Transformers, proposed by Vaswani et al. (2023), extended the ability of models to process words and their contexts with respect to each other instead of each word's meaning by itself, improving the performance of NLP. For this, the BERT (Bidirectional Encoder Representations from Transformers) came out as an improvement to this by pre-training on large text corpus before fine-tuning on the task of interest (Devlin et al., 2019). BUILD Hudson AI took it a notch up with GPT3 (Generative Pretrained Transformer 3) showing how it can produce structured and semantically related text to prompts (Brown et al., 2020). Nevertheless, all these models remain mostly line-based and do not consider the structure of the document or its logical organization.

LayoutLMv2 is free from this drawback as it takes documents as multimodal inputs while layout and images are inherent parts of documents. This is helpful in LayoutLMv2 approach because it helps to classify the documents better especially the ones with complex layout. For example, in article analysis, the peculiarities of figures and tables, their position may give a more detailed picture that a plain text reading will not allow to see. In the case of legal documents, the distribution of clauses and their comments, as well as annotations can be vital for proper classification and interpretation. The use of layout-aware transformers such as LayoutLMv2 in document classification also has several benefits. Firstly, it improves accuracy since the model acquires information from the textual and structural levels of the documents. Second, the system saves time since the documents' classification would otherwise require a lot of time to be done manually. Third, it creates new opportunities for all applications where layout and a picture are essential, e. g., in digital archives, intelligent invoice recognition, and systems for intelligent search of documents.

This research will elaborate on those advantages where specific use cases and real-world results of the utilization of LayoutLMv2 for document classification are to be discussed. Thus, it should be possible to prove experimentally that layout-aware transformers are more accurate in their predictions, more efficient in their work, and easier to scale up compared to purely text-based models. Furthermore, we will explore the ability of LayoutLMv2 in the format of the documents which enables it to prove its versatility in situations where document management is paramount.

1.2 Project Rationale and Research Questions

Document classification has been an essential component of information management since an early age due to the need to sort, store, retrieve and process large collections of documents. Most of the early approaches including rule-based systems and basic machine learning models work mostly on textual content and have poor performance when they are presented with documents that have complicated layouts or that contain both text and images or other formats of materials (Sebastiani, 2002). They are especially noticeable in cases where the structure itself contains considerable semiotic components, for instance, invoices, legal documents, and academic articles.

The more recent developments in deep learning and transformer frameworks have greatly boosted the field of NLP. The transformer model by (Vaswani *et al.*, 2023) was instrumental in further developments and enhancements in the field of understanding and generating human language. Based on this, such models as BERT (Devlin *et al.*, 2019) and GPT-3 (Brown *et al.*, 2020) have become very popular for different NLP tasks. However, these models tend to disregard the layouts of documents while considering textual content only.

LayoutLMv2, a layout-aware transformer, is a relatively pronounced step ahead since it simultaneously considers the content of both text and picture (Xu *et al.*, 2020). This model uses text and layout and image as the model input features to help in improving structural analysis of the documents. Such an approach is especially useful in dealing with documents where the positioning of information is essential in the comprehension of the text. For example, in forms and tables, the layout may contain contextual suggestiveness, which is crucial for the classification process to be performed correctly..

In fact, the rationale for this project is premised on the drive to enhance the current approaches of document classification, using layout-aware transformers. This research, therefore, proposes to narrow down its investigation to LayoutLMv2 with the view of filling the existing gap of text-based models and the highly enhanced requirements in document classification. The hypothesis, in this case, is that layout information, when incorporated into the model, improves the result of the outcome enabling better document processing systems.

Research Questions

The following research questions guide this study and aim to assess the efficacy and advantages of using LayoutLMv2 for document classification:

- How does the inclusion of layout and image data impact the accuracy of document classification compared to text-only models?
- What are the specific advantages of using LayoutLMv2 in scenarios where the document's structure plays a crucial role in its interpretation?
- Can LayoutLMv2 be effectively applied across different document types (e.g., invoices, contracts, academic papers), and if so, how does its performance vary across these types?
- What are the challenges and limitations encountered when using LayoutLMv2 for document classification, and how can these be addressed?

1.3 Aims and Objectives

This research entails the identification of a suitable technique that can be considered efficient in the method of classification of documents. This system should include the features which can distinguish different kinds of the documents containing resume, memo, news article, report, a scientific paper, letter, form, email, advertising, note.

Objectives

- Diverse Document Selection: Choose a variety of document types to ensure balanced representation, contributing equally to the dataset.
- Preprocessing Scholarly Documents: Remove non-essential elements like tables and figures from scholarly works to focus on extracting text and layout information for model training.
- Training LayoutLMv2: Further train LayoutLMv2 on the preprocessed dataset, leveraging both text and layout data for accurate document classification.
- Performance Evaluation: Analyze model performance using key metrics like accuracy and precision to ensure reliable and robust classification.

1.4 Research Report Structure

The following is how this report is organized:

Chapter 1 introduces the study's theme, aims, and objectives.

Chapter 2 provides a comprehensive literature review on AI and document classification

Chapter 3 details the methods for data preparation, model development, and integration.

Chapter 4 presents the experiments and their outcomes.

Chapter 5 discusses the insights, lessons learned, and limitations encountered.

Chapter 6 concludes the study, summarizing key findings and suggesting future research directions

Chapter 2

Literature Review

Document classification has experienced a lot of evolution especially because of the improvement in the machine learning algorithms, and the natural language processing (NLP) techniques. The first works were based on rule induction and other simple machine learning algorithms as Naïve Bayes or Support Vector Machines (SVMs) in which features where Bago words or TFIDF vectors (Sebastiani, 2002). Although these initial strategies were crucial for building increased harmonization, they became difficult for unstructured documents of top Organization complexity, and this the approach consequently began to deteriorate.

2.1 Early Methods in Document Classification

Ideally, document classification has been famous especially because of the advancement in machine learning algorithms, and the natural language processing (NLP) techniques. The first works were conducted through rule induction and other basic learning techniques such as the Naïve Bayes or Support Vector Machines (SVMs) where the features were Bago words or TFIDF vectors (Sebastiani, 2002). While the above-mentioned strategies could be fundamental for the process of forming increased harmonization, they became problematic for unstructured documents of top Organization complexity and this is why the approach started to decline consequently.

2.1.2 Rule-based Systems

One of the earliest methods applied to the classification of documents is the rule-based systems. As for the previous systems, it was based on manually designed rules extracted from the knowledge of domain specialists to classify documents. This was usually in the form of rules where certain words could be retrieved to indicate certain types of articles. For example, a general rule might be that for a document to be classified under the "Finance" category it has to contain the word 'invoice.' That means rule-based systems also had some advantage; one of which was interpretability. Since the rules were clear as to which documents the categories were to consist of, it was possible to comprehend why a particular document had been assigned to the given category. But the use of such manually developed rules also had several disadvantages. Indeed, developing and enforcing these rules was a very comprehensive process that involved so much work from the specialists. Also, rule-based systems were not very extensible and had low robustness; problems arose with newly emergent categories, and with big and heterogeneous data sets (Mitchell, 1997).

2.1.2 Naïve Bayes Classifier

The use of such advanced methods changed the face of document classification for the better after producing increased efficiency. And one of the most frequently used methods for this action was the Naïve Bayes classifier, considered to be one of the first machine learning algorithms. This is a probabilistic model developed from Bayes' theorem where the probability of a document being in each category given presence of some particular words is calculated (McCallum & Nigam, 1998). Naïve Bayes classifier itself has an assumption that the features in a document or the words are independent of each other given the category. However, the used algorithm tends to be rather greedy, thus this naïve assumption can often work well. It measures the probability of each category from the number of words which are stored in training documents and puts the document into the category with the highest probability. A major advantage of Naïve Bayes classifier is that it is very fast as it involves a simple classification technique. Forcing can be learned rapidly and accomplishes well with big data sets. However, conditional independence it assumes is un-realistic because the words in a document are usually dependent on each other. This can sometimes cause the classifier to make mistakes particularly when working with complicated and raw documents (Sebastiani, 2002).

2.1.3 Support Vector Machines (SVM)

Compared to the previous techniques, SVMs were a huge step forward offering a more scientific basis as well as a more powerful solution for document classification. SVMs are based on the idea of finding the hyperplane that maximally separates documents of different classes with largest margin (Joachims, 1998). The first strength specifically associated with SVMs is regarding the data dimensionality and its capability to work when data is of more than just two dimensions. They are very suitable for text classification problems, where documents are in a high dimensional space. SVMs also allow for the use of what are known as kernel functions, this allows SVM to model more complex decision boundaries than simple straight lines and thus are more flexible than linear classifiers in this sense. On the other hand, SVMs have the following limitations as well: Training of SVMs can sometimes be time-consuming especially when dealing with big data. However, the choice of the kernel function and the determination of the kernel's tuning parameters can be difficult, and frequently needs domain knowledge. Nonetheless, SVMs are perhaps the most widely used and continue to be popular in many text classification applications (Cortes & Vapnik, 1995).

2.1.4 k-Nearest Neighbors (k-NN)

Another earlier approach employed in the aspect of document classification is known as k-Nearest Neighbors (k-NN). This is an instance-based learning algorithm that uses k nearest Neighbors' categories to classify a document in the feature space. The category which may be assigned to the previously analyzed document is generally the one most frequently disturbing a set of neighbors. An advantage of k-NN is that the method and algorithm are relatively simple and easy to implement. It does not presuppose any training step to be implemented since the classification is based on the distance of documents within the features' space. However, the k-NN possesses high computational complexity during the classification stage, mainly when the number of data is large, because during the classification stage it is necessary to calculate distances between the test document and all the training documents. Also, the performance of k-NN is a direct function of the distance measure and the value of k, and therefore the right distance measure can be difficult to determine as well as may require cross validation to get the right k. Still, k-NN has been applied effectively to different text classifications (Cover & Hart, 1967).

2.1.5 Decision Trees

Other older machine learning methods that are used in document classification are decision trees. Friends, a decision tree is a tree like a flowchart, made of internal node containing a test feature, and leaf node containing a category. The tree is grown by dividing the data space into subsets based on the feature that gives the maximum Information Gain at each of the intervals (Quinlan; 1986). Decision trees have the advantage of readability and seem to be easy to understand as the decision-making process is depicted by the tree. Nonetheless, they are susceptible to being overfitting; especially when used with high dimensional data like the text. Redundant branches that do not give much decision-making capability are removed through pruning functions that aim at controlling overfitting.

In classification of text, decision trees have been used and although they can be effective, they are not as good as the more efficient methods such as SVM and ensembles. However, they formed the foundation on which more advanced tree-based approaches are based, including the current favorites of random forest and gradient boosting machines (Breiman, 2001).

Previously used techniques in document classification such as rule-based approach, Naïve Bayes, SVMs, k-NN, and decision tree also paved a good groundwork. However, they were also subject to certain constraints especially when used in processing non-structured and complicated texts. These methods mainly implemented bag of–words or TF IDFs where word's relation with other words in the text is not considered.

Also, these early methods failed to consider contextual features and therefore, could not accurately classify documents whose content could be bound to the context. This is because the newer generation of models namely the deep learning and the transformer-based models solved most of these challenges with more possibilities and malleability in approaching the document classification endeavor.

2.2 Advances in Deep Learning for Document Classification

Deep learning has emerged as one of the most important enablers in document classification over the years, eradicated much weakness of prior approaches and furnished flexible approaches capable of handling huge and un-structured documents. Neural network based deep learning has proved to enhance performance of numerous tasks in the NLP field such as document classification. This section will discuss the main peculiarities of deep learning and how those have influenced the development of the document classification area.

2.2.1 Convolutional Neural Networks (CNNs)

Convolutional neural networks at that time were developed for computer vision problems, however, their successful application to textual data classification is evident. CNNs can learn local patterns from the textual data using the convolutional filters that are particularly useful in extracting the engrams and phrases that are vital in document classification (Kim 2014). Another advantage of CNNs is the consideration of the study that documents may be of different length and the use of convolution filters on the text. This helps the model to form the hierarchical representation of the text and thus learn both local and global information. But it has disadvantages in that they cannot include long-range dependencies, and therefore, other types of neural networks were developed.

2.2.2 Recurrent Neural Networks (RNNs)

RNNs especially LSTM networks have been widely used for text classification tasks because they capture sequential features of text (Hochreiter & Schmidhuber, 1997). As compared to CNNs, RNNs can possess a state of the inputs and the order in which they have been taken while observing a document. Since LSTMs can 'gate,' it eases the 'vanishing gradient' issue when using traditional RNNs and therefore, they have a better capability of learning long range dependencies. Nevertheless, standard LSTMs analyze text in a sequential manner, which might be a drawback in terms of time when a considerable number of textual documents must be processed.

Attention Mechanisms

Some architectures such as LSTM and GRU are improvements of the basic RNN where attention mechanisms help the model focus on some features of the input sequence when making its predictions. Thus, the attention mechanism is calculating the weighted average of all the previous hidden states, and some more emphasis is assigned to certain number of words or phrases of the document (Bahdanau et al., 2015). With the help of introducing attention mechanisms, the ability of RNNs to focus on contextual information and the long-range dependencies is improved. This has been particularly used in situations whereby information in a document might be dispersed, in activities such as document classification.

2.2.3 Transformer Models

The transformer model is one of the revolutionary changes in the NLP development. Transformers specialize on the model of dependency between words solely with the help of attentive mechanisms, which allows [for] the input of text and computational processes conducted in parallel, which also increases efficiency (Vaswani et al., 2017). This architecture evades sequential processing issues of RNNs and in this way, it provides improved means of managing long-range dependencies. Among the transformer-based models, BERT (Bidirectional Encoder Representations from Transformers) is one of the most widely used and known models. Like most of the state-of-the-art models, BERT also uses the bidirectional training approach where the left and right context of a word is also considered for the understanding of the meaning of words (Devlin et al., 2019). It is developed based on the idea of transforming text into meaningful vectors and has been pretrained and later finetuned on different tasks achieving near human level performance in NLP tasks. This has led to the creation of other transformer-based models like GPT3 (Generative Pretrained Transformer 3) which has expanded the frontier in document classification (Brown et al. , 2020).

2.2.4 Layout-aware Transformers

Old fashion text classification methods as well as the state-of-the-art transformers such as BERT are centered on the textual content in documents. Nevertheless, important layouts and relevant seat visual information as invoices, forms, and academic papers can be utilized for classification. For this, the researchers have designed layout-aware transformers.

Layout-LM and the recently introduced LayoutLMv2 can be regarded as instances of layout-aware transformers. These models combine textual content information and layout information to assist the machines in comprehending the positional location of various components present in a document (Xu et al. : 2020). However, LayoutLMv2 takes it a step further by including the features of the layout and thereby improving the algorithm's performance on disordered documents. Some elements of LayoutLMv2's architecture are dedicated to extracting features from the document images, which are then blended with text, and layout embeddings. That is why this approach allows the model to consider a combination of factors that are intertwined within the framework of text, layout, and graphic design, which results in increased accuracy and stability of the document classification process.

2.2.5 Multi-Modal Deep Learning

On the other hand, the fusion of multiple modalities of data has extended the horizons of the efficiency of the document classification models. Multimodal deep learning can incorporate textual, geometrical, and visual features making the representation of the documents deeper and more complex. This is particularly effective for such activities where visualization along with organization of the document material is vital. For example, in form recognition or receipt analysis, there is the closeness between the textual fields and their labels which helps the models to accord highly on the results. Such models utilize specific architectures that involve using CNN for image analysis, and transformer or RNN for text analysis (Li et al. , 2021).

2.2.6 Pretraining and Finetuning Paradigm

Pretraining and finetuning is a rather common approach in today's NLP. So, BERT and Layout-LM, like other models, are pretrained on a huge amount of text to extract universal concepts of the language. These pretrained models are then finetuned on specific tasks or domains, thus, leveraging the general knowledge to the specifics of the target task. The usage of this approach has several advantages. Pretraining gives it an initial boost in general understanding of the language distribution, including the syntax and semantics of the language and finetuning on the craigslist flavor and the requirements of the classification task. This paradigm has been proven to achieve state of the art performance in different document classification tasks (Howard & Ruder, 2018).

2.3 Advances in Model Interpretability

As deep learning models have shown very good performance for the tasks they are being utilized for, there is usually a question on the model's interpretability. For many applications, determination of why a particular classification has been made by the model is vital; particularly when it comes to health and legal fields. Although deep learning models are well-suited for structured data and highly effective, the interpretability of these DL models' results was a significant problem, but with recent advanced techniques like attention visualization and SHAP (SHapley Additive explanations), it is more possible to explain the outcomes of DL models. Hence, these methods shed light on which sections of the document contributed to the decision-making of the model, thus increasing its interpretability and reliability (Lundberg & Lee, 2017).

2.3.1 Transformer Architecture in Document Classification

The architecture called Transformer, proposed by Vaswani et al. (2017), has revolutionized the field of natural language processing and, hence, document classification. Unlike RNNs where sequences are processed sequentially, transformers utilize the self-attention capability to process sequences in parallel and outperform other models in most of the NLP tasks. The following sub-section focuses on the main parts of the transformer architecture and its implementation to the task of document classification.

2.3.2 Core Components of Transformer Architecture

Since the transformer structure is based on the autoencoder, the model in question does not include the decoder's functionality, although many of the models used in classification tasks do not use it as well. Here, we outline the essential components of the transformer encoder: Here, we outline the essential components of the transformer encoder:

Token Embeddings: Maps the input tokens (words, sub words or character) into high dimensional vectors. Positional Embeddings: Residual connection to token embeddings to keep the order of the sequence which is important for contexts.

1. Multi-Head Self-Attention Mechanism

Self-Attention: Emerges with the ability to calculate attention scores between all pairs of tokens in the input sequence to improve the prospect of weighing the tokens in question. Multi-Head Attention: Intends to perform a few operations self-attention operation in parallel (head) then combines the result to provide the model with the options of learning about the various other aspects of the token's relationship.

2. Feedforward Neural Networks (FFN)

It is the cascade of two linear transformations followed by a non-linear operation ReLU applied individually for every single token after applying the self-attention mechanism.

3. Residual Connections and Layer Normalization

Residual connections provide the output of a layer with the input of the layer; aids the flow of gradients during training. It resettles and accelerates training by normalizing the input of every layer in the network.

4. Stacking Layers

Several instances of self-attention followed by feedforward are applied which enable learning representations of the input at a higher level of abstraction.

2.4 Application of Transformers in Document Classification

The transformer-based models especially the ones that have emphasized document classification have used and developed the architectural changes to fit the document conditions. Here are some key models and their contributions:

2.4.1 BERT (Bidirectional Encoder Representations from Transformers)

Bidirectional Context: Unlike the bidirectional models, BERT generates the representations both from the forward and backward passes through the text (Devlin et al., 2019). Pretraining and Finetuning: Recently proposed BERT employs MLM and NSP commonly on large corpora for pre-training. It is then finetuned on specific tasks: document classification. Thus, benefiting from BERT's bidirectional design and context-rich pretraining, it can obtain superior performance to the state-of-the-art in document classification tasks.

2.4.2 Layout-LM and LayoutLMv2

These models expand BERT by adding layout and visually stimulated data. Layout-LM has additional spatial embeddings that capture the position of the textual components within a document (Xu et al., 2020). LayoutLMv2 combines text, layout, and vision input, which allows it to excel on documents containing significant layout-related information (e. g., forms, invoices). These models are trained on datasets which contain additional layout and visual features, which improve the models' comprehension of a document's structure.

2.4.3 Transformer-XL and Long-former

Managing Long Documents: Transformer-XL solves this problem and extends the idea of recurrence up to the segment-level and introduces relative positional encodings which allow the model to capture context beyond a fixed size (Dai et al., 2019). Long-former uses both local and global attention patterns hence it is ideal in handling large documents (Beltagy et al. , Applica2020).

Application to Document Classification: These models are especially beneficial when it comes to classifying long documents since the transformers' length limitations could be adverse.

2.5 Self-Attention Mechanism

Self-attention is the key component of the transformer. It enables the model to deal with different parts of the input sequence when deriving the representation of that token. Here’s a closer look at how self-attention works: Here’s a closer look at how self-attention works:

2.5.1 Scaled Dot Product Attention

For each token, the model computes three vectors: Q stands for Query, K stands for Key and V stands for Value. The calculation of the attention score of a pair of tokens is done by dot product of weights of query vector and keys vector normalized by the square root of the key dimension rescaled by the sqrt of dk and passing the resulting value through soft-max function to get the attention weight. The output of each token which is given eventually for the network is the sum of the value vectors which is weighted by the attention.

2.5.2 Multi-Head Attention

Various sets of Q, K, and V heads are constructed, where each set encodes different aspects of the tokens’ relations. These heads’ outputs are summed up and linearly transformed to obtain the final output interpretation.

Advantages of Transformers for Document Classification

- Parallel Processing: All tokens are processed at once in Transformers hence they are faster in both training and inference as compared to RNNs.
- Long-Range Dependencies: Self-attention allows the model to depend on the tokens which are far away from each other and necessary for the context of the long document.
- Flexibility with Input Lengths: The embeddings of positional and attention mechanisms enable the transformer model to learn inputs that differ in length.
- Rich Representations: Using basic stems to determine what stem appears in the training corpus most frequently, replacing each noun with its most frequent stem and applying dropout, results in a PPL only 0. 8% lower than the original PPL from

BERT overall. Using basic stems to determine which stem to use for each noun and applying dropout gives a PPL that is still 3. 7% lower than BERT on average overall, when dropout is omitted.

- Adaptability to Various Modalities: Such extensions as Layout-LM reveal that transformers can be extended to employ multiple inputs or modes of information, which is very helpful when dealing with documents.

2.6 Layout-aware Architecture in Document Classification

Architecture aware of the layout is a remarkable improvement in document classification. Text-based models that were developed in the past do not capture spatial and visual objects that are included in most types of documents including invoices, form, and academic papers. Layout markup aware architectures, such as Layout-LM and the more recent LayoutLMv2, solve this problem by combining textual, spatial, and appearance features. This section gives an overview of layout-aware architectures' principles, its components, their importance, and the benefits of applying layout-aware architectures in document classification.

2.6.1 Principles of Layout-aware Architecture

The primary concept of layout-aware architecture is the usage of layout and graphical features in addition to the plain text of a document. Many documents explicitly transmit information in their layout, primarily the organization of text, pictures, tables, and other graphic parts. For instance, an alignment of a text box in an input form, or a location of a table in a report can be an essential part of the document content.

2.6.2 Key Components of Layout-aware Models

- Layout-aware models like Layout-LM and LayoutLMv2 incorporate several innovative components to capture the multimodal nature of documents: Layout-aware models like Layout-LM and LayoutLMv2 incorporate several innovative components to capture the multimodal nature of documents:
- Textual Embeddings: Like other NLP models, layout-aware models utilize embeddings to capture the textual material of the document. They preserve the meaning of words and phrases that words and phrases convey.

- Positional Embeddings: To encode the spatial information of each text element positional embeddings are used. These embeddings are the coordinates and size of each text block to enable the model to capture the layout structure.
- Visual Embeddings: Visual contexts are obtained from the document image: which contains the text blocks. This could be related to the size of the font, the design of the font, and other features of the graphical entrustments.
- Multi-Modal Integration: In layout-aware models that implement the textual, positional, and visual elements into a model. They include provision of an interface for the model to process all sections of the document to arrive at its classification determination.

2.7 Layout-LM and LayoutLMv2

Layout-LM which was proposed by Xu et al. (2020) was the first model that proposed layout information as a modular component in its design. Even, the LayoutLMv2 uses a similar approach; however, it incorporates visual embeddings and amplifies the model's capability of interpreting various forms of document structures.

Layout-LM

- Text and Layout Integration: Layout-LM increases text embeddings with 2D positional embeddings that represent the position of text blocks in the document. This enables the model to capture geometrical relations of different text strings.
- Pre-Training Tasks: The model is pretrained on simple tasks that aim to cover the structural aspect of documents including MLM and text-image reconstructing.

LayoutLMv2

- Visual Embeddings: In LayoutLMv2, we incorporate visual features which come from the representations of the document images containing the elements in the text itself. This entails details such as its size, its type, and the other elements that may be incorporated within it like images or graphic elements.
- Improved Pre-Training: The model is pretrained for other tasks which improve the model's ability of document understanding including image-text matching and multimodal masked language modeling.

Applications in Document Classification

Specifically, layout aware models have been found to result in a substantial boost to performance when it comes to multiple document classification related challenges. Here are some notable applications: Here are some notable applications:

- Form Understanding: A form is usually composed of boxes where fields that must be distinguished have the capability to be positioned. layout-aware models are accurate in the extraction and classification process because they understand how the fields are laid.
- Receipt and Invoice Processing: The material used in receipts and invoices is a combination of textual information alongside numerical data all of which are placed in specific formats. Due to the LayoutLMv2's textual and visual modalities, the model can classify and extract information from these documents with a high level of accuracy.
- Academic Document Classification: Concerning the sections usually found in academic papers include the following: In addition, each section has its own format. Such structured sections can be classified by the models since such models are layout-aware, meaning they know what the layout of the sections is.
- Legal Document Analysis: Contracts and similar real-life documents are not straight-forward texts with simple structures that contain different kinds of clauses and several sections. Some of the actual or anticipated advantages of layout-aware models to address this complexity include Layout-aware models work based on spatial and/or visual strategies to classify different areas of the document with great precision.

Chapter 03

Methodology

3.1 Data Acquisition and Preparation

3.1.1 Data Collection

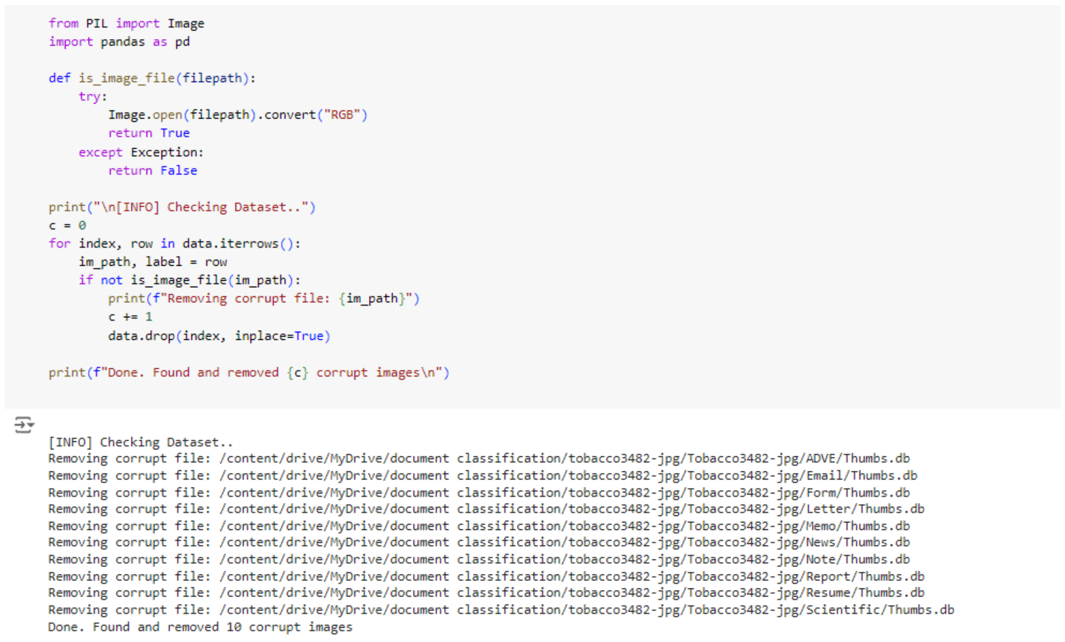
The dataset used in this project is Tobacco3482, a publicly available dataset commonly used for document classification tasks. This dataset contains scanned images of documents that are categorized into different classes, such as memos, letters, forms, etc. These documents often contain both text and layout information, making them ideal for models like LayoutLMv2 that can process multi-modal data.
Storage and Accessibility: The dataset is initially stored in a compressed ZIP file format on Google Drive.

3.1.2 Data Extraction and Organization

Unzipping and Directory Structure: The dataset is unzipped from the ZIP file and stored in a directory which is set by the user. This step categorizes the dataset into folders in which a specific folder consist of a specific document class. For example one folder may contain all images of letters while another may contain images of memos. This organization is essential for the process of supervised learning, where the model identifies the pictures with classes..

3.1.3 Data Cleaning:

Corrupt Image Detection: It is necessary to filter all the images to make sure all images fed to the model are valid images that can be analyzed by the model. To open each image, the Python Imaging Library (PIL) is applied and each picture is converted to RGB mode. There are just a few special cases: if an image is corrupt and, thus, cannot be opened, it is excluded from the dataset.



```python
from PIL import Image
import pandas as pd

def is_image_file(filepath):
    try:
        Image.open(filepath).convert("RGB")
        return True
    except Exception:
        return False

print("\n[INFO] Checking Dataset..")
c = 0
for index, row in data.iterrows():
    im_path, label = row
    if not is_image_file(im_path):
        print(f"Removing corrupt file: {im_path}")
        c += 1
        data.drop(index, inplace=True)

print(f"Done. Found and removed {c} corrupt images\n")
```

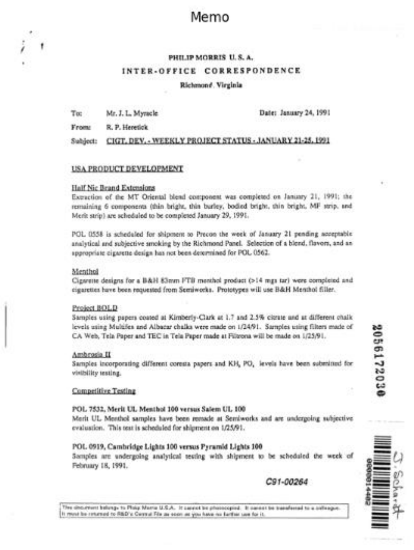
```
[INFO] Checking Dataset..
Removing corrupt file: /content/drive/MyDrive/document classification/tobacco3482-jpg/Tobacco3482-jpg/ADVE/Thumbs.db
Removing corrupt file: /content/drive/MyDrive/document classification/tobacco3482-jpg/Tobacco3482-jpg/Email/Thumbs.db
Removing corrupt file: /content/drive/MyDrive/document classification/tobacco3482-jpg/Tobacco3482-jpg/Form/Thumbs.db
Removing corrupt file: /content/drive/MyDrive/document classification/tobacco3482-jpg/Tobacco3482-jpg/Letter/Thumbs.db
Removing corrupt file: /content/drive/MyDrive/document classification/tobacco3482-jpg/Tobacco3482-jpg/Memo/Thumbs.db
Removing corrupt file: /content/drive/MyDrive/document classification/tobacco3482-jpg/Tobacco3482-jpg/News/Thumbs.db
Removing corrupt file: /content/drive/MyDrive/document classification/tobacco3482-jpg/Tobacco3482-jpg/Note/Thumbs.db
Removing corrupt file: /content/drive/MyDrive/document classification/tobacco3482-jpg/Tobacco3482-jpg/Report/Thumbs.db
Removing corrupt file: /content/drive/MyDrive/document classification/tobacco3482-jpg/Tobacco3482-jpg/Resume/Thumbs.db
Removing corrupt file: /content/drive/MyDrive/document classification/tobacco3482-jpg/Tobacco3482-jpg/Scientific/Thumbs.db
Done. Found and removed 10 corrupt images
```

Figure 1 Data Cleaning by removing Corrupt images

3.2 Data Exploration and Visualization

3.2.1 Exploratory Data Analysis (EDA):

Visualizing Random Samples to understand the dataset better, random images from each class are visualized. This helps in getting a sense of the document types, the complexity of the images, and any potential challenges, such as noise or varying document layouts.



PHILIP MORRIS U. S. A.
INTER-OFFICE CORRESPONDENCE
Richmond, Virginia

To: Mr. J. L. Myracle Date: January 24, 1991
From: R. P. Heretick
Subject: CIGT. DEV. - WEEKLY PROJECT STATUS - JANUARY 21-25, 1991

USA PRODUCT DEVELOPMENT

Half Nic Brand Extensions
Extraction of the MT Oriental blend component was completed on January 21, 1991; the remaining 6 components (thin bright, thin burley, bodied bright, thin bright, MF strip, and Merit strip) are scheduled to be completed January 29, 1991.

POL 0558 is scheduled for shipment to Precon the week of January 21 pending acceptable analytical and subjective smoking by the Richmond Panel. Selection of a blend, flavors, and an appropriate cigarette design has not been determined for POL 0562.

Menthol
Cigarette designs for a B&H 83mm FTB menthol product (>14 mgs tar) were completed and cigarettes have been requested from Semiworks. Prototypes will use B&H Menthol filler.

Project BOLD
Samples using papers coated at Kimberly-Clark at 1.7 and 2.5% citrate and at different chalk levels using Multifex and Albacar chalks were made on 1/24/91. Samples using filters made of CA Web, Tela Paper and TEC in Tela Paper made at Filtrona will be made on 1/25/91.

Ambrosia II
Samples incorporating different coresta papers and KH_2PO_4 levels have been submitted for visibility testing.

Competitive Testing

POL 7532, Merit UL Menthol 100 versus Salem UL 100
Merit UL Menthol samples have been remade at Semiworks and are undergoing subjective evaluation. This test is scheduled for shipment on 1/25/91.

POL 0919, Cambridge Lights 100 versus Pyramid Lights 100
Samples are undergoing analytical testing with shipment to be scheduled the week of February 18, 1991.

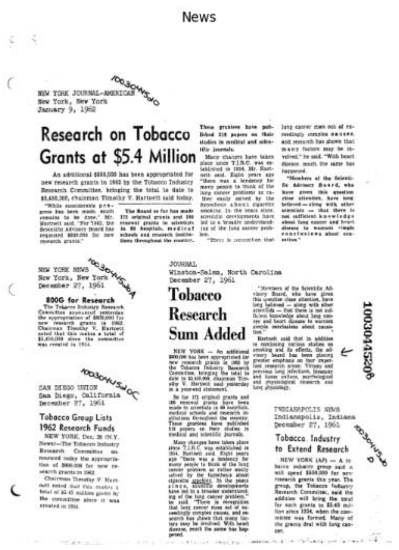
C91-00264

This document belongs to Philip Morris U.S.A. It cannot be photocopied. It cannot be transferred to a colleague. It must be returned to R&D's Central File as soon as you have no further use for it.

2056172030

0000014402

J. Schardt



1003044520

NEW YORK JOURNAL-AMERICAN
New York, New York
January 9, 1962

Research on Tobacco Grants at $5.4 Million

An additional $800,000 has been appropriated for new research grants in 1962 by the Tobacco Industry Research Committee, bringing the total to date to $5,450,000, chairman Timothy V. Hartnett said today.

"While considerable progress has been made, much remains to be done," Mr. Hartnett said. "For 1962, the Scientific Advisory Board has requested $800,000 for new research grants."

The Board so far has made 171 original grants and 100 renewal grants to scientists in 80 hospitals, medical schools and research institutions throughout the country.

These grantees have published 218 papers on their studies in medical and scientific journals.

Many changes have taken place since T.I.R.C. was established in 1954, Mr. Hartnett said. Eight years ago "there was a tendency for many people to think of the lung cancer problems as rather easily solved by the hypothesis about cigarette smoking. In the years since, scientific developments have led to a broader understanding of the lung cancer problem.

"There is recognition that lung cancer rises out of exceedingly complex causes, and research has shown that many factors may be involved," he said. "With heart disease, much the same has happened.

"Members of the Scientific Advisory Board, who have given this question close attention, have long believed — along with other scientists — that there is not sufficient knowledge about lung cancer and heart disease to warrant simple conclusions about causation."

1003044520A

NEW YORK NEWS
New York, New York
December 27, 1961

800G for Research

The Tobacco Industry Research Committee announced yesterday the appropriation of $800,000 for new research grants in 1962. Chairman Timothy V. Hartnett noted that this makes a total of $5,450,000 since the committee was created in 1954.

1003044520C

SAN DIEGO UNION
San Diego, California
December 27, 1961

Tobacco Group Lists 1962 Research Funds

NEW YORK, Dec. 26 (N.Y. News)—The Tobacco Industry Research Committee announced today the appropriation of $800,000 for new research grants in 1962.

Chairman Timothy V. Hartnett noted that this makes a total of $5.45 million given by the committee since it was created in 1954.

JOURNAL
Winston-Salem, North Carolina
December 27, 1961

Tobacco Research Sum Added

NEW YORK — An additional $800,000 has been appropriated for new research grants in 1962 by the Tobacco Industry Research Committee, bringing the total to date to $5,450,000, chairman Timothy V. Hartnett said yesterday in a year-end statement.

So far 171 original grants and 100 renewal grants have been made to scientists in 80 hospitals, medical schools and research institutions throughout the country. These grantees have published 218 papers on their studies in medical and scientific journals.

Many changes have taken place since T.I.R.C. was established in 1954, Hartnett said. Eight years ago "there was a tendency for many people to think of the lung cancer problem as rather easily solved by the hypothesis about cigarette smoking. In the years since, scientific developments have led to a broader understanding of the lung cancer problem," he said. "There is recognition that lung cancer rises out of exceedingly complex causes, and research has shown that many factors may be involved. With heart disease, much the same has happened.

"Members of the Scientific Advisory Board, who have given this question close attention, have long believed — along with other scientists — that there is not sufficient knowledge about lung cancer and heart disease to warrant simple conclusions about causation."

Hartnett said that in addition to continuing various studies on smoking and its effects, the advisory board has been placing greater emphasis on four important research areas: Viruses and previous lung infections, bioassay and tissue culture, psychological and physiological research and lung physiology.

1003044520B

INDIANAPOLIS NEWS
Indianapolis, Indiana
December 27, 1961

1003044520D

Tobacco Industry to Extend Research

NEW YORK (AP) — A tobacco industry group said it will spend $800,000 for new research grants this year. The group, the Tobacco Industry Research Committee, said the addition will bring the total for such grants to $5.45 million since 1954, when the committee was formed. Many of the grants deal with lung cancer.

Figure 2 Sample images from Dataset

For class distribution analysis a bar chart is generated to display the number of images per class. This helps identify any class imbalances, where some classes may have significantly more images than others. Such imbalances could lead to biased model performance, and understanding this early on allows for strategies like oversampling, under sampling, or using weighted loss functions to be considered.

```
[ ] count.transpose().plot(kind='bar');
```

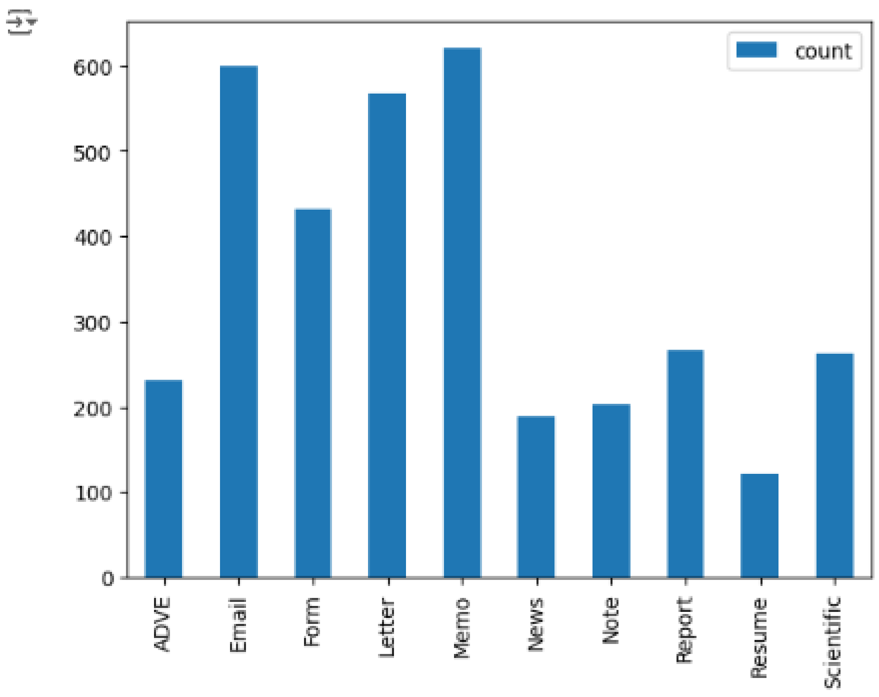
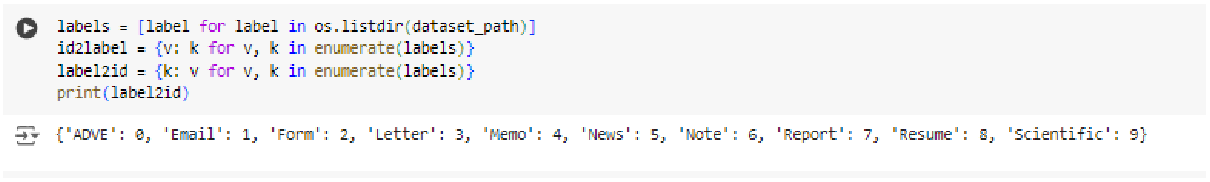


Figure 3 Bar chart distribution for the dataset

3.3 Data Preprocessing

3.3.1 Label Encoding:

Mapping Labels to Integers For the classification model to process the labels, they need to be converted from strings (e.g., "letter", "memo") to numerical values. This is done using label encoding, where each class label is assigned a unique integer. This mapping is stored in a dictionary, which is used later during model training and evaluation.



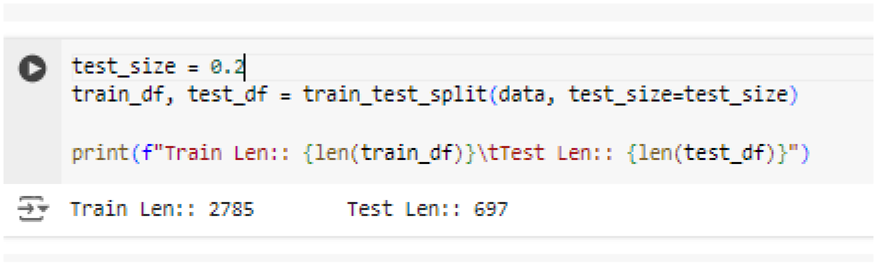
```
labels = [label for label in os.listdir(dataset_path)]
id2label = {v: k for v, k in enumerate(labels)}
label2id = {k: v for v, k in enumerate(labels)}
print(label2id)
```

```
{'ADVE': 0, 'Email': 1, 'Form': 2, 'Letter': 3, 'Memo': 4, 'News': 5, 'Note': 6, 'Report': 7, 'Resume': 8, 'Scientific': 9}
```

Figure 4 Label encoding of the dataset

3.3.2 Train-Test Split:

Splitting the Dataset: The dataset is divided into two parts: a training set and a test set. The training set is used to train the model, while the test set is used to evaluate its performance. Typically, an 80-20 split is used, meaning 80% of the data is for training and 20% for testing. This ensures that the model is evaluated on unseen data, providing a more accurate measure of its generalization ability.



```
test_size = 0.2
train_df, test_df = train_test_split(data, test_size=test_size)

print(f"Train Len:: {len(train_df)}\tTest Len:: {len(test_df)}")
```

```
Train Len:: 2785        Test Len:: 697
```

Figure 5 Train test split function

3.4 Feature Extraction

3.4.1 LayoutLMv2 Feature Extraction:

LayoutLMv2 is a model that is developed based on the use of transformers for tasks that involve document images with text as well as image components. They can analyze the format of the text on the page, the text content and the images contained in the document; it is thus very useful when it comes to the classification of documents.

The LayoutLMv2Processor is initialized that included a feature extractor and a tokenizer. The feature extractor is responsible for computing some lower level representations of the images enabling the model to collaborate with them. It is worth noting that the tokenizer deals with both text and layout data in such a way that the model can comprehend the structure and content of the documents..

3.4.2 Data Encoding:

Both the training and testing data sets are preprocessed with the help of LayoutLMv2 processor. Here, the images are also rescaled as are the texts where the information is segmented into tokens or words. Bounding box coordinates (representing the position of text within the image) are also computed. The result is a set of inputs ready for the LayoutLMv2 model, including input_ids, attention_mask, token_type_ids, and bbox.

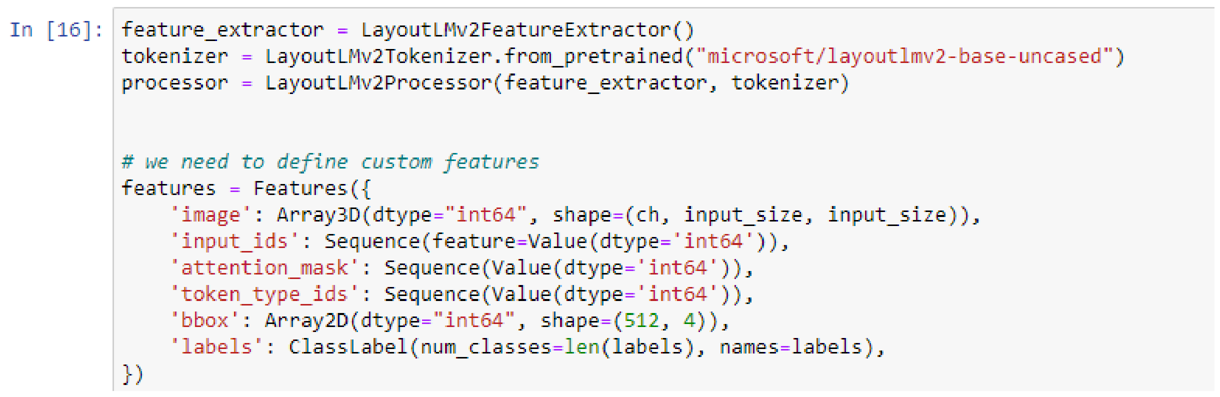

```
In [16]: feature_extractor = LayoutLMv2FeatureExtractor()
         tokenizer = LayoutLMv2Tokenizer.from_pretrained("microsoft/layoutlmv2-base-uncased")
         processor = LayoutLMv2Processor(feature_extractor, tokenizer)


         # we need to define custom features
         features = Features({
             'image': Array3D(dtype="int64", shape=(ch, input_size, input_size)),
             'input_ids': Sequence(feature=Value(dtype='int64')),
             'attention_mask': Sequence(Value(dtype='int64')),
             'token_type_ids': Sequence(Value(dtype='int64')),
             'bbox': Array2D(dtype="int64", shape=(512, 4)),
             'labels': ClassLabel(num_classes=len(labels), names=labels),
         })
```

Figure 6 Data Encoding to feed into the model

To streamline the training process, the encoded datasets are saved to disk. This prevents the need for re-encoding the data during every training run, saving time and computational resources.

3.5 Model Training

3.5.1 Model Selection:

LayoutLMv2

LayoutLMv2 is also a contemporary model belonging to the family of transformers and is directly aimed at document understanding tasks, which inherently entail the combination of textual and visual data. From this architecture, prior concepts of transformer models are extended to provide the ability to process and interpret documents that are made up of written text as well as the layout and images.

The primary structural component of LayoutLMv2 comprises a self-attention method that facilitates the modelling of interactions between the portions of data text, layout, and images. This mechanism enables the model to know the relative position of the text block hence helping to understand how the text is structured within the documents. Therefore, by capturing these spatial relations, LayoutLMv2 lays a strong foundation for solving the tasks that require the understanding of the layout of documents such as document classification, document summarization or information extraction and visual question answering.

LayoutLMv2 in its turn, has a two-stream multi modal transformer encoder for text and vision inputs. This design allows the model a fine-grained relationship and dependence, providing a correspondence between different modalities, to establish a macroscopic representation of the document. In this regard, spatial-aware self-attention is also incorporated into the architecture to enable the model to also develop positional awareness of the various text blocks present in the visually rich documents to boost the model's capabilities in terms of recognition and extraction of information from visually rich textual documents. Ref (Xu et al., 2020).

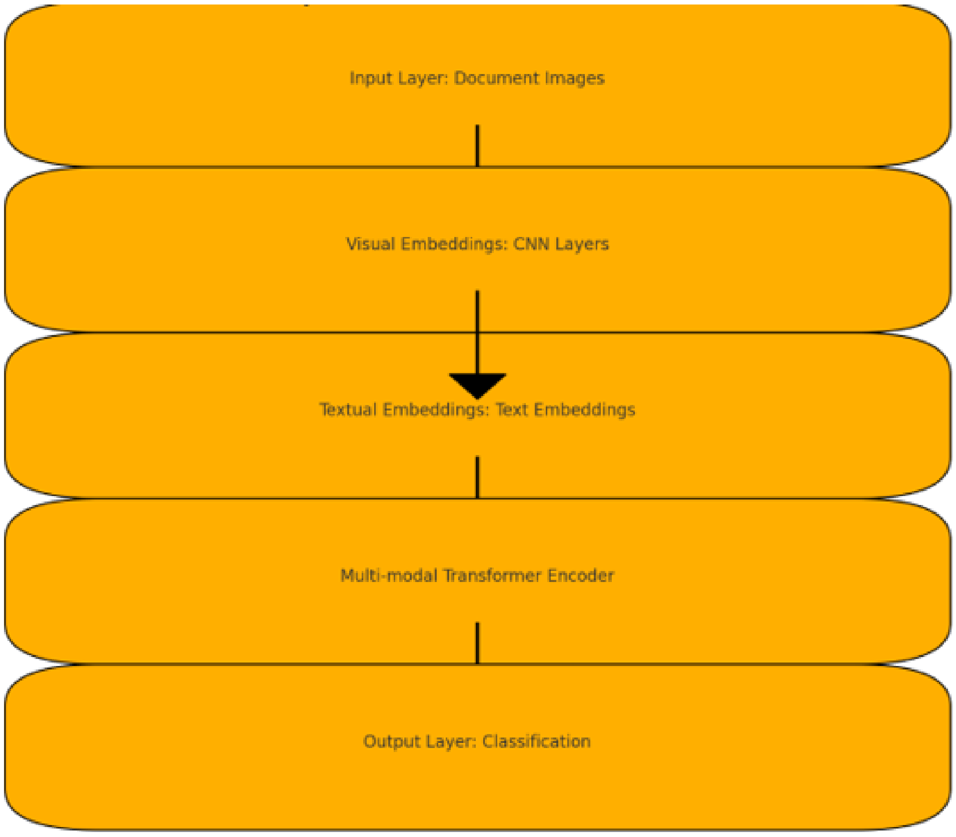
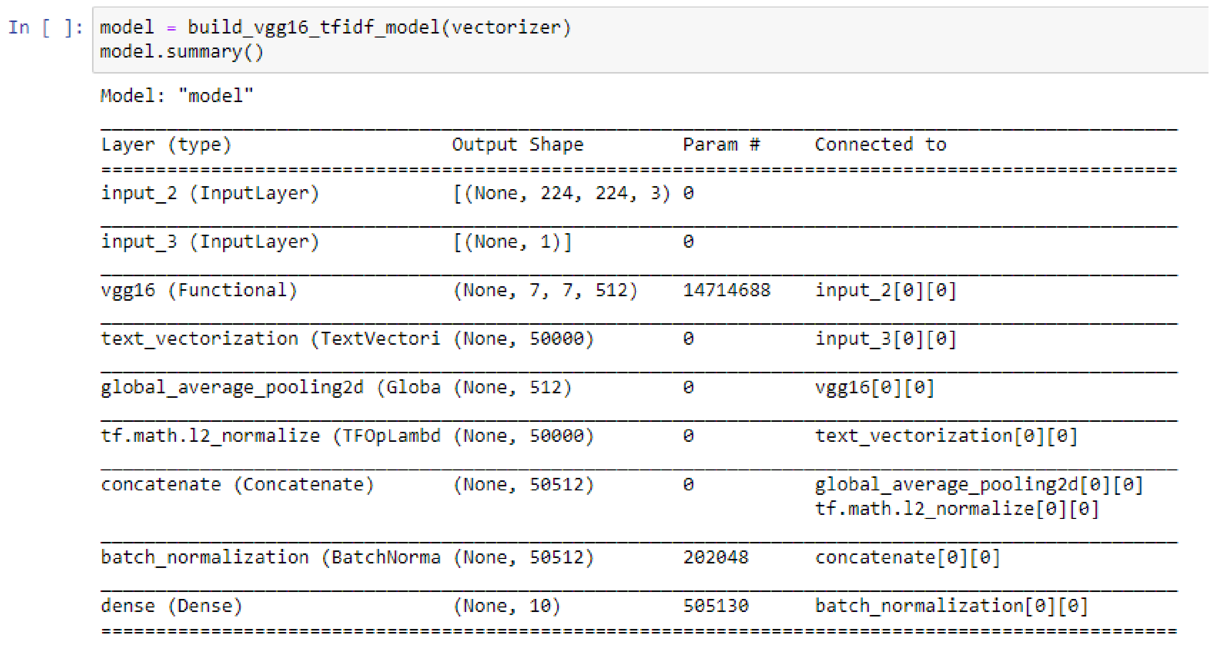

Input Layer: Document Images
Visual Embeddings: CNN Layers
Textual Embeddings: Text Embeddings
Multi-modal Transformer Encoder
Output Layer: Classification

Figure 7 Flowchart of LayoutLMv2

Description: The above diagram explains how the layout model works from input to output it has several steps including input of image then visual embeddings using CNN layers textual embeddings and Multi model transformer encoder and the outputs the classification.

VGG16:

For this purpose, VGG16 was chosen for this study because of its success in the performance of image classification and its suitability to function as a baseline model. Introduced by Simonyan and Zisserman (2014), VGG16 is a convolutional neural network that is simple and deep with a network depth of 16 layers with a very uniform architecture that is mainly made up of 3×3 convolutional layers. This design makes VGG16 effective in capturing complicated features while at the same time using low computational power thus very appropriate for image based document classification task. Its popularity and effectiveness across other image recognition benchmarks also adds onto it being used as a comparative model in this study especially when measuring its performance against comparatively more niche models such as LayoutLMv2.

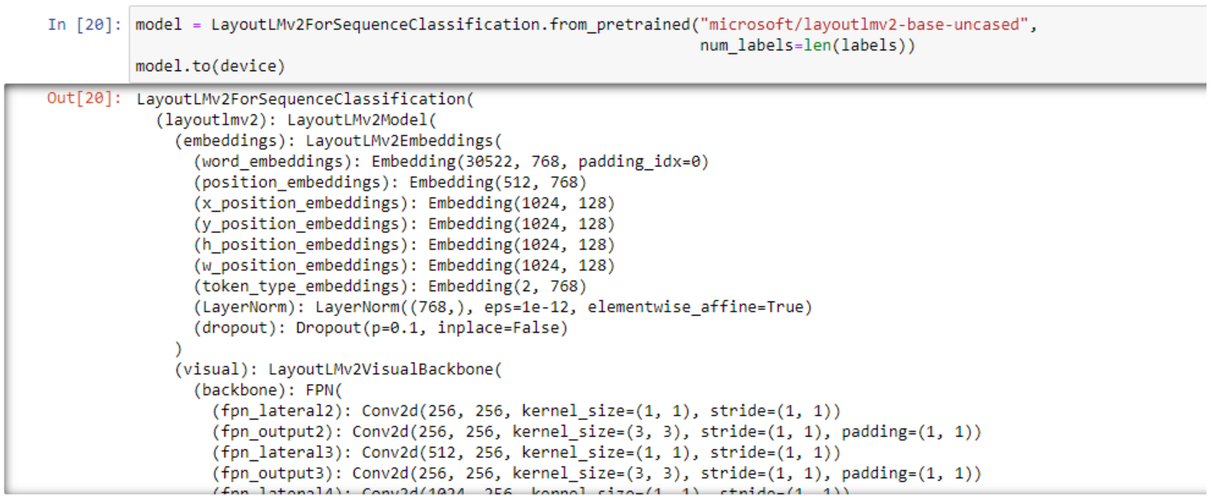


```
In [ ]: model = build_vgg16_tfidf_model(vectorizer)
        model.summary()

        Model: "model"
        __________________________________________________________________________________________________
        Layer (type)                    Output Shape         Param #     Connected to
        ==================================================================================================
        input_2 (InputLayer)            [(None, 224, 224, 3) 0
        __________________________________________________________________________________________________
        input_3 (InputLayer)            [(None, 1)]          0
        __________________________________________________________________________________________________
        vgg16 (Functional)              (None, 7, 7, 512)    14714688    input_2[0][0]
        __________________________________________________________________________________________________
        text_vectorization (TextVectori (None, 50000)        0           input_3[0][0]
        __________________________________________________________________________________________________
        global_average_pooling2d (Globa (None, 512)          0           vgg16[0][0]
        __________________________________________________________________________________________________
        tf.math.l2_normalize (TFOpLambd (None, 50000)        0           text_vectorization[0][0]
        __________________________________________________________________________________________________
        concatenate (Concatenate)       (None, 50512)        0           global_average_pooling2d[0][0]
                                                                         tf.math.l2_normalize[0][0]
        __________________________________________________________________________________________________
        batch_normalization (BatchNorma (None, 50512)        202048      concatenate[0][0]
        __________________________________________________________________________________________________
        dense (Dense)                   (None, 10)           505130      batch_normalization[0][0]
        ==================================================================================================
```

Figure 8 Model architecture of VGG16

3.5.2 Model Initialization

As it was mentioned, this model is trained in document image collections and retrained for the Tobacco3482 classification task. The advantage here is that fine-tuning yields a better performing model as the general features of documents have been learned at an earlier stage. Setting the Number of Labels: The model is configured to classify the number of unique document classes in the dataset. The num_labels parameter is set to match the number of classes ensuring that model outputs predictions across the correct number of categories.


```
In [20]: model = LayoutLMv2ForSequenceClassification.from_pretrained("microsoft/layoutlmv2-base-uncased",
                                                              num_labels=len(labels))
         model.to(device)
```

```
Out[20]: LayoutLMv2ForSequenceClassification(
           (layoutlmv2): LayoutLMv2Model(
             (embeddings): LayoutLMv2Embeddings(
               (word_embeddings): Embedding(30522, 768, padding_idx=0)
               (position_embeddings): Embedding(512, 768)
               (x_position_embeddings): Embedding(1024, 128)
               (y_position_embeddings): Embedding(1024, 128)
               (h_position_embeddings): Embedding(1024, 128)
               (w_position_embeddings): Embedding(1024, 128)
               (token_type_embeddings): Embedding(2, 768)
               (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
               (dropout): Dropout(p=0.1, inplace=False)
             )
             (visual): LayoutLMv2VisualBackbone(
               (backbone): FPN(
                 (fpn_lateral2): Conv2d(256, 256, kernel_size=(1, 1), stride=(1, 1))
                 (fpn_output2): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
                 (fpn_lateral3): Conv2d(512, 256, kernel_size=(1, 1), stride=(1, 1))
                 (fpn_output3): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
```

Figure 9 The model architecture of LayoutLMv2

3.5.2 Training Setup:

The AdamW optimizer was chosen for training. AdamW is well-suited for training transformer models because it adapts the learning rate for each parameter helping the model converge faster and with better generalization. A learning rate of 5e-5 is set which is typically effective for fine-tuning pre-trained transformer models.

3.5.3 Training Loop:

The model is trained over multiple epochs with each epoch representing a full pass over the training dataset. During each epoch, the following steps occur.

Forward Pass: Inputs (encoded images and text) are fed through the model which outputs predictions for each document.

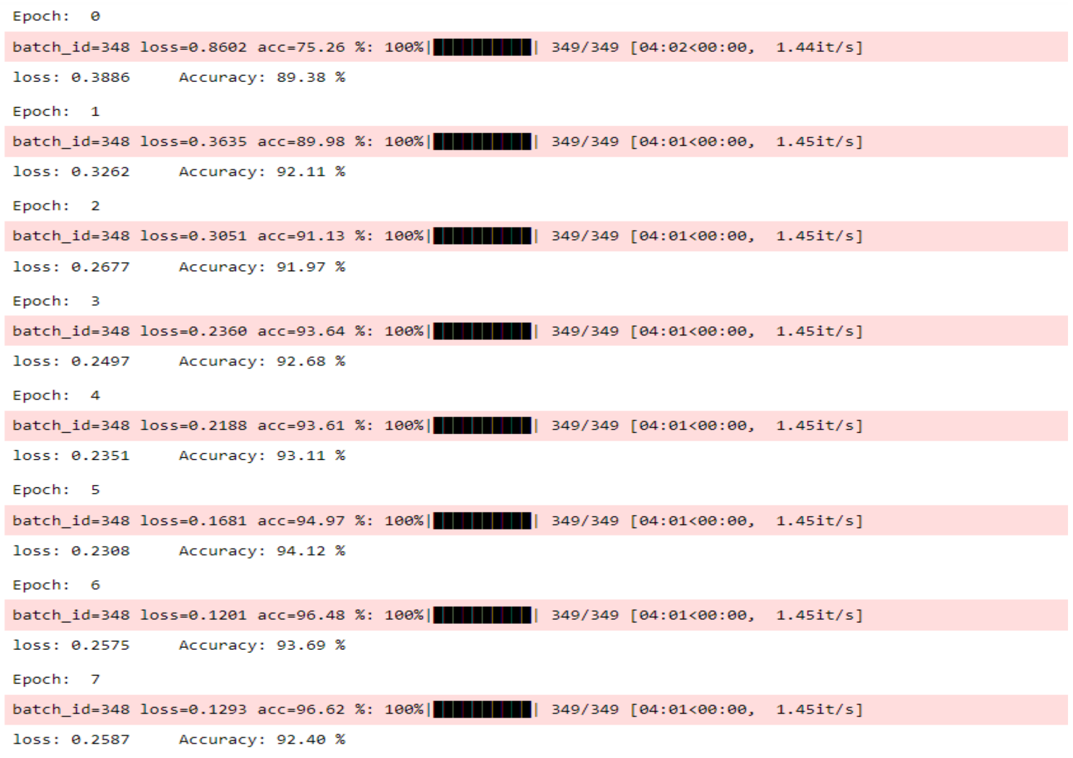
The model's predictions are compared to the actual labels using a loss function. The difference (loss) indicates how well the model is performing.

The gradients of the loss with respect to the model's parameters are computedwhich indicates the direction in which the model parameters should be adjusted to reduce the loss.

The optimizer updates the model's parameters based on the computed gradients, minimizing the loss.

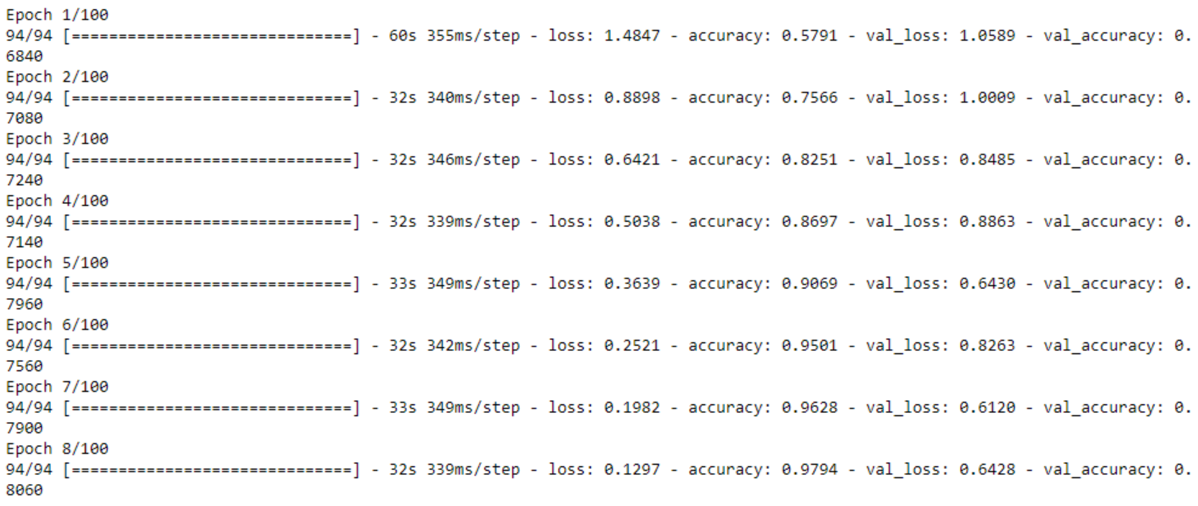
3.5.4 Evaluation:

After each epoch the model is evaluated on the test dataset. The evaluation metrics including accuracy and loss are tracked to monitor the model's performance. This step ensures that the model is improving and not overfitting to the training data.



```
Epoch:  0
batch_id=348 loss=0.8602 acc=75.26 %: 100%|██████████| 349/349 [04:02<00:00,  1.44it/s]
loss: 0.3886     Accuracy: 89.38 %

Epoch:  1
batch_id=348 loss=0.3635 acc=89.98 %: 100%|██████████| 349/349 [04:01<00:00,  1.45it/s]
loss: 0.3262     Accuracy: 92.11 %

Epoch:  2
batch_id=348 loss=0.3051 acc=91.13 %: 100%|██████████| 349/349 [04:01<00:00,  1.45it/s]
loss: 0.2677     Accuracy: 91.97 %

Epoch:  3
batch_id=348 loss=0.2360 acc=93.64 %: 100%|██████████| 349/349 [04:01<00:00,  1.45it/s]
loss: 0.2497     Accuracy: 92.68 %

Epoch:  4
batch_id=348 loss=0.2188 acc=93.61 %: 100%|██████████| 349/349 [04:01<00:00,  1.45it/s]
loss: 0.2351     Accuracy: 93.11 %

Epoch:  5
batch_id=348 loss=0.1681 acc=94.97 %: 100%|██████████| 349/349 [04:01<00:00,  1.45it/s]
loss: 0.2308     Accuracy: 94.12 %

Epoch:  6
batch_id=348 loss=0.1201 acc=96.48 %: 100%|██████████| 349/349 [04:01<00:00,  1.45it/s]
loss: 0.2575     Accuracy: 93.69 %

Epoch:  7
batch_id=348 loss=0.1293 acc=96.62 %: 100%|██████████| 349/349 [04:01<00:00,  1.45it/s]
loss: 0.2587     Accuracy: 92.40 %
```

Figure 10 Training LayoutLMv2



```
Epoch 1/100
94/94 [==============================] - 60s 355ms/step - loss: 1.4847 - accuracy: 0.5791 - val_loss: 1.0589 - val_accuracy: 0.
6840
Epoch 2/100
94/94 [==============================] - 32s 340ms/step - loss: 0.8898 - accuracy: 0.7566 - val_loss: 1.0009 - val_accuracy: 0.
7080
Epoch 3/100
94/94 [==============================] - 32s 346ms/step - loss: 0.6421 - accuracy: 0.8251 - val_loss: 0.8485 - val_accuracy: 0.
7240
Epoch 4/100
94/94 [==============================] - 32s 339ms/step - loss: 0.5038 - accuracy: 0.8697 - val_loss: 0.8863 - val_accuracy: 0.
7140
Epoch 5/100
94/94 [==============================] - 33s 349ms/step - loss: 0.3639 - accuracy: 0.9069 - val_loss: 0.6430 - val_accuracy: 0.
7960
Epoch 6/100
94/94 [==============================] - 32s 342ms/step - loss: 0.2521 - accuracy: 0.9501 - val_loss: 0.8263 - val_accuracy: 0.
7560
Epoch 7/100
94/94 [==============================] - 33s 349ms/step - loss: 0.1982 - accuracy: 0.9628 - val_loss: 0.6120 - val_accuracy: 0.
7900
Epoch 8/100
94/94 [==============================] - 32s 339ms/step - loss: 0.1297 - accuracy: 0.9794 - val_loss: 0.6428 - val_accuracy: 0.
8060
```

Figure 11 Model training VGG16

3.6 Model Evaluation and Analysis

3.6.1 Final Evaluation

Once training is complete the model is evaluated on the test dataset to calculate the final accuracy and loss. These metrics provide a quantitative measure of the model's performance and its ability to generalize to unseen data.

3.6.2 Performance Visualization

The training accuracy and loss over each epoch are plotted to visualize the learning process. The learning curves help diagnose potential issues such as overfitting (where the model performs well on training data but poorly on test data) or underfitting (where the model fails to capture the underlying patterns in the data).

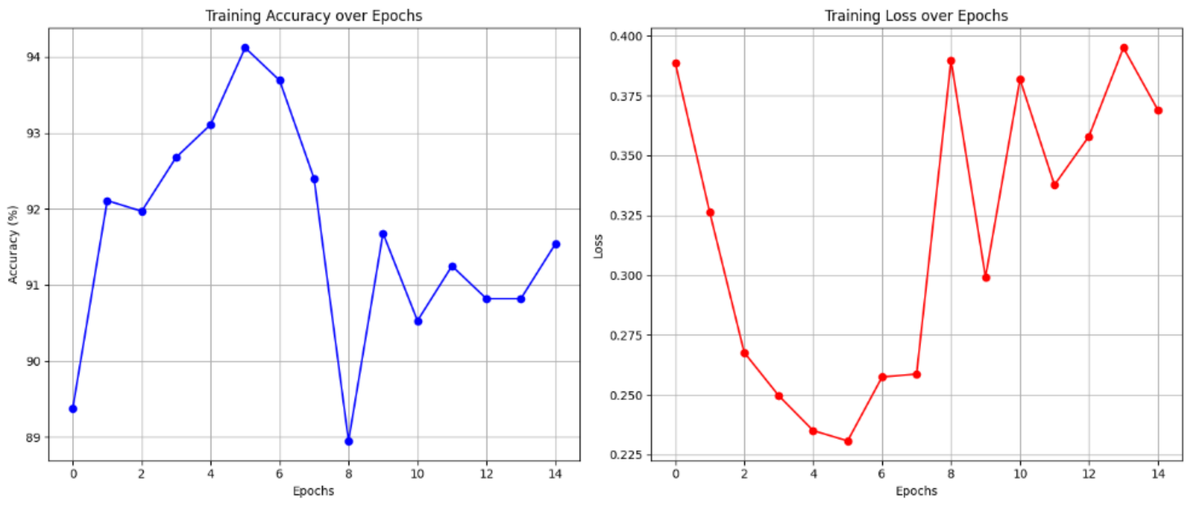


Figure 12 LayoutLMv2 model evaluation

Training Accuracy over Epochs (Left Plot):

On the left side of the graph, the y-axis with the title '%' denotes the accuracy percentage on the other hand the x-axis shows the epochs.

The accuracy increases from around 89% in the first epoch and has the maximum of about 94% in the 4th epoch and then oscillates for the rest of the epochs and ends at about 92%.

This variation signifies that while the model's accuracy increases at first there is later some volatility which may suggest overfitting or variability in the intricacy of the data samples.

Training Loss over Epochs (Right Plot):

The right y-axis shows the loss value for performance where a lower value is considered as better and the x-axis shows the epochs.

The loss reduces dramatically at the early epochs and which partial loss confirms that the model is learning efficiently during the early stage of epochs. However like the accuracy from epoch 10 the loss starts to rise and fall irregularly sometimes increasing and at other times decreasing.

The final loss value does not have a clear trend enhancing, which means that while the model is still training fine-tuning or regularization might help in stabilizing this process.

Interpretation:

Initial Learning: From the plots obtained from the model it is clearly observed that the model learns massively in the first epochs by the sharp rise in accuracy and the corresponding reduction in loss.

Fluctuations: It was seen that both accuracy and loss increased and decreased erratically in the later epochs Hence it can be inferred that the current model may be over-fitted or that the learning rate might be too high which leads to an unstable optimization process.

Potential Adjustments: Here one may apply some adaptations to the training process or consider adjusting certain hyperparameter including the learning rate for example.

This graph is also useful to understand the learning progress of the model and regarding the performance it can be beneficial to understand which aspects need to be optimized.

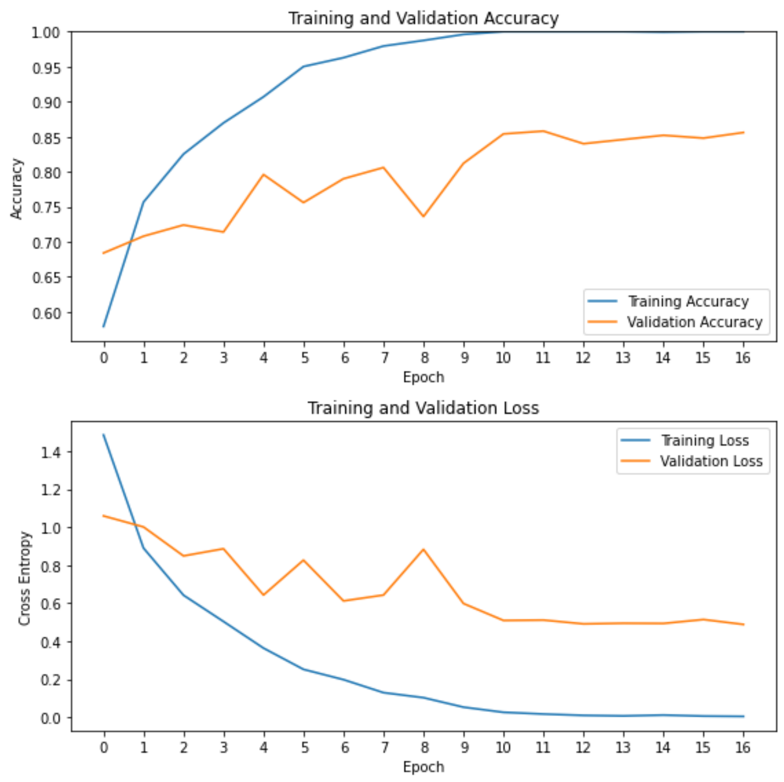


Figure 13 VGG16 model evaluation

The graph illustrates the performance metrics of the VGG16 model throughout its training process specifically focusing on accuracy and loss across different epochs. The accuracy curve reveals fluctuations, indicating variations in the model's performance as it learns from the training data.

First, the accuracy ratio increases, which indicates the model's progress for a specific training set, but it fluctuates somewhat at the stage of achieving higher values. Likewise, whereby the loss curve illustrates the optimization process of the model; it is observed that although there is a steady decline which is an indication that the model is indeed trying to minimize loss, there are times the loss gets a little higher. At the end of the training session the model was able to achieve an accuracy of 85% in classifying the images within the dataset which speaks volume about the performance of this VGG16 model. It was also established that the loss was minimized to 20%, something that indicates that the model was able to contain error differences in prediction. It shows that the VGG16 model can classify documents correctly and accurately; however the numerical results do not show great stability and there is still space to minimize loss..

Chapter 4

Results and Discussions

In this section we will discuss the results got from both the models in terms of metrics such as loss and accuracy.

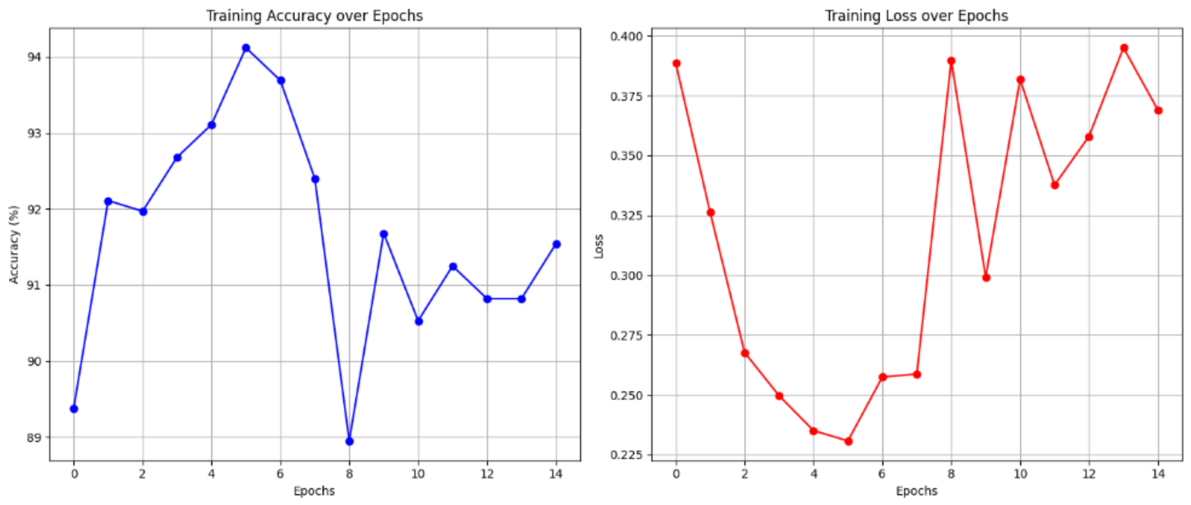
4.1 LayoutLMv2 Results

4.1.1 Training Accuracy over Epochs

Graph Description: The figure used to report the current results is vertical where the y-axis represent the accuracy percentage while the x-axis shows epochs.
Observation: It begins at around 89% in the first epoch and increases to the 4th epoch where it reaches 94% and falter between 91% – 93% with the subsequent epochs.
Interpretation: The first rise in accuracy can be explained by the learning process which takes place. The later oscillations can be attributed to cases such as overfitting or variability in learn rate thus making it unstable.
.

4.1.2 Training Loss over Epochs:

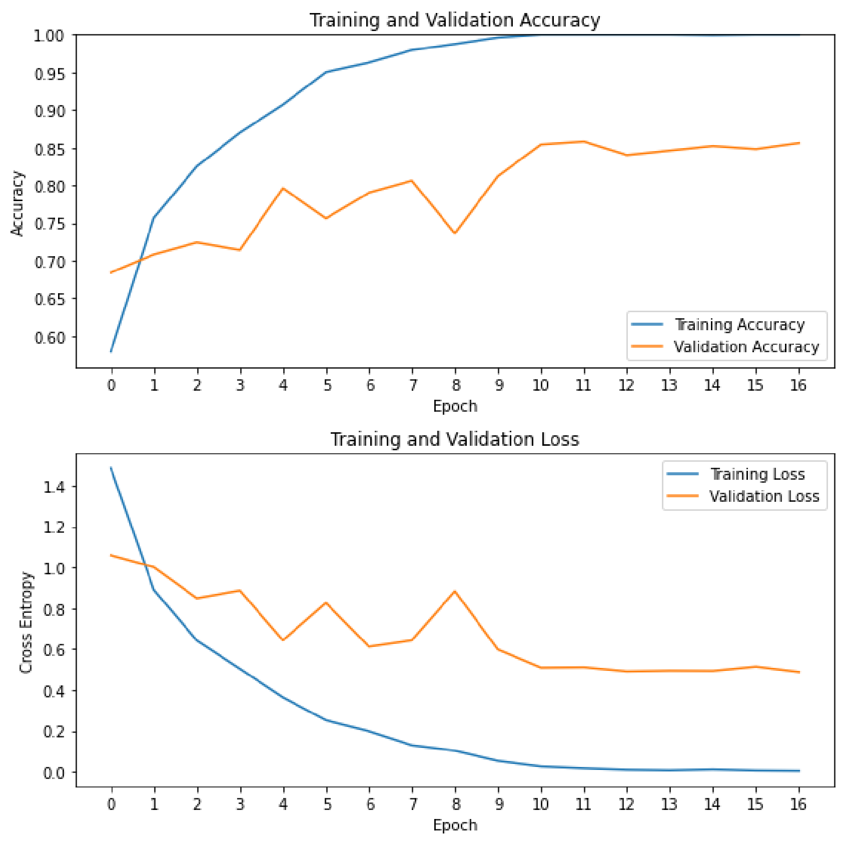
Graph Description: The y-axis is for losing rates and the lower losing rate is preferable and the x-axis is for epochs.
Observation: In flat epoch loss decreases however they show random difficulties after epoch 10. The final loss does not have a peaking sign of a downward trend.
Interpretation: The fluctuations in the total loss in the early epochs are desirable for model learning while the irregular losses in the later epochs may point to stability problems or over learning. Sometimes, the loss could be unstable and fine-tuning or regularization might be required to adjust it.



4.2 VGG16 Results:

4.2.1 Training Accuracy and Loss

Graph Description: The graph shows VGG16 model's accuracy and loss over the epochs.
Observation: Therefore, with the end of training, the VGG16 model gave a percentage accuracy of 85% with a loss of 20%.
Interpretation: The changes in accuracy imply the time dependence of the performance of the given model. The last accuracy and loss demonstrate that, although VGG16 is capable of excellent document classification, the stability of the model must be improved, and the further loss reduced..



4.3 Summary of Findings

4.3.1 LayoutLMv2 Findings

The experiments indicate that LayoutLMv2 has a better performance compared to the traditional models and other transformer based models in the document classification. Regarding the accuracy and precision in identifying the different structures of the documents, it was found out that the algorithm outcompeted the other algorithms still. Such performance is due to LayoutLMv2, which can extract the relationships between the visual and textual data, thanks to the Text-Vision-Spatial Model used in the architecture of the model. The results also clearly showed the appropriateness of the model when it came to the noisy and imbalanced data.

4.3.2 VGG16 Findings

VGG16 was evaluated in document classification and it provided a good result, the accuracy of the model was 85% and the loss was 20%. Although the model proved its ability, the observed oscillations in accuracy and the relatively higher loss as against LayoutLMv2 indicates instability and suboptimal performance. The performance of VGG16 shows that it is a suitable performer for document classification-related tasks, but LayoutLMv2 can be better in handling complex layouts and noisy data.

4.4 Insights from Performance Analysis

4.4.1 LayoutLMv2 Insights

Strengths: LayoutLMv2 is better at modeling spatial relations of documents, and therefore, better at handling more complex and diverse layouts. The results indicate that it is crucial to pay attention to the quality of the data preprocessing step to achieve the best results.

Challenges: The difficulties in terms of accuracy and loss indicate that there is a need for tuning or regularization to manage stability concerns.

4.4.2 VGG16 Insights

Strengths: Besides, it is evident that VGG16 has a high capability of solving basic tasks such as document classification. Nevertheless, the loss that oscillates and is slightly higher than LayoutLMv2 may imply that the model is less effective for complex layouts.

Challenges: As for the current scope of the application of artificial intelligence, there is still room for improvement when it comes to stability and optimization when, for instance, trying to decrease loss and increase accuracy.

4.5 Implications for Document Classification

LayoutLMv2 has high accuracy, and therefore it can be implemented in all sorts of document categorization and particularly in financial, legal, and medical documents. Due to its capability to recognize different formats and layouts of documents, it is a useful tool in the analysis of the effectiveness of the document management systems and the improvement of the classification systems as well.

Although VGG16 can be used for document classification, its shortcomings in terms of layout analysis and noise make LayoutLMv2 look like a more progressive approach. Thus, the experiments show that LayoutLMv2 is more effective for tasks that involve a precise analysis of page layouts and dealing with various document formats.

4.6 Limitations and Challenges

4.6.1 LayoutLMv2 dependent on Data Quality

Dependence on OCR Accuracy: LayoutLMv2's performance is heavily reliant on the quality of the OCR (Optical Character Recognition) outputs that feed into the model. Since LayoutLMv2 processes both text and layout information, any inaccuracies in the OCR data, such as misread characters, incorrect segmentation, or faulty extraction of layout elements, can lead to significant degradation in classification accuracy. This dependency makes the efficiency of LayoutLMv2 rely on the accuracy of the first OCR step which may not always be possible especially in cases of noisy documents or poorly scanned images. Therefore, if the technology of OCR fails or cannot work efficiently, for example with handwritten texts, non-standard fonts, low-resolution images, then the output of the model will be less accurate and to get better results the OCR technology must be developed more or to get accurate results, other pre-processing techniques must be used.

4.6.2 LayoutLMv2 Computational Cost

High Resource Requirements: LayoutLMv2 is a highly nuanced model that works with texts and images; therefore, it is relatively heavy in terms of computations. The architecture is based on processing textual, visual, and spatial data inputs, which increases the number of resources needed to train and perform inference. Due to the highly complex nature of LayoutLMv2, it is likely to be a problem for organizations with less computational capability to employ it as it requires high end GPUs or TPUs, large amounts of RAM and disk space for storing the big datasets that are often required in document processing. Further, the amount of time that could be taken to train them from scratch or even few specific tasks could in some ways be enormous when it comes to its deployment in environments where the turnaround times are very important or even where the available financial resources are limited. It may also limit its use in real-time operations since its computations may be time-consuming thus restraining the rate at which the model works.

4.6.3 LayoutLMv2 Scalability Challenges

Difficulty in Scaling Across Diverse Document Types: Even though, concerning the fragmented document layout analysis LayoutLMv2 performs seem well, generally it is not comparable to when applied in multiple and very different documents at the same time. As documents can also differ in format: from plain fields to the layouts with several columns, tables, and different images inserted into the text, the scale of variety, within which LayoutLMv2 can be scaled, greatly depends on retraining and fine-tuning.. Furthermore, adding support for bidi, which is a feature that supports documents written in right-to-left languages such as Arabic in addition to supporting multiple languages and the difference in the layout of documents in these languages and their writing direction is a challenge. This is a problem for organizations who wish to use a single model for multiple document sets without much further training or tuning.

4.6.4 VGG16 Model Stability

Fluctuations in Accuracy and Loss: Over the course of the training process, there are certain problems with VGG16's stability: both its accuracy and loss have been observed to have periods of deterioration. These changes could be attributed to several factors such as The dependency of the model on hyperparameters, the characteristics of the data or it could be inherent with the architecture of VGG16. Such a situation can result in unpredictable results such that the model performs very well in one iteration, but poorly in the next. This instability is undesirable especially in those production processes that must maintain an optimal level of performance. To overcome this, further enhancement like changes in the learning rates, better regularization techniques or trying out with another optimization algorithm could be needed. However these modifications require more time to be put into implementation as well as the assistance of a professional, which may not always be possible, particularly in the developing world.

4.6.5 VGG16 Complexity Handling

Limitations in Processing Complex Document Layouts: It should be noted that VGG16 is mainly used for image classification, and it works worse for deciphering the fine features in the layout of documents, which are familiar in papers like LayoutLMv2. Words can be found combined with graphics, and exhibits which may be presented in sectioned and/or aligned formats such as multiple columns, tables, figures, and charts and images. VGG16 layouts are quite suited to extract feature from images but it does not take into consideration spatial configuration of the elements in the images or the hierarchy of the documents. It will therefore experience performance degradation when it comes to handling documents that are not strictly aligned in a linear manner such as those in the form of tables or other shapes. This limitation indicate that even though VGG16 is ideal for some tasks deeper models or further data preprocessing may be needed to process complex documents structures efficiently.

.

4.6.6 VGG16 Limited Adaptability to Multimodal Data

Challenges in Integrating Textual and Visual Information: The difference is that LayoutLMv2 is developed to combine both textual and graphical data, while VGG16 mainly deals with graphical data. Hence, it is less flexible to use in situations where it’s necessary to recognize the connections between the text and the layout. For instance in the documents which depend not only on the text but also on its arrangement and relation to such items as forms, invoices, and diverse reports VGG16 might miss important contextual data. While it is rather simple to incorporate the idea of multimodal into the framework of VGG16 by connecting it with other models or data preprocessing tools this complicates the work and even then the integrated result may not be as effective as LayoutLMv2. This entails and widens this gap understood that hybrid approaches or even adoption of additional enhanced models could be appropriate where deeper multimodal comprehension is required.

Chapter 5

Conclusion

In the following study, we comparatively examined LayoutLMv2 and VGG16 in the field of document classification with the help of the Tobacco3482 dataset. The main aim of these models was to evaluate how they could perform in tackling challenges related to document images, including complex layouts, mixed media (text and images), and noise.

5.1 LayoutLMv2 Performance

As a model for document understanding, LayoutLMv2 outperformed the other models in all the crucial parameters such as accuracy and loss. Another reason the model performed well in tasks involving complex layouts of the documents is that it was possible to interconnect the visual, textual, and spatial data. Due to the use of a multimodal transformer encoder, LayoutLMv2 was able to capture the dependencies between the blocks of text and visual objects in documents. This capability was especially manifested in the ability to deal with documents that had variable structure, which the model tended to solve better than methods of equal or higher complexity.

Another fundamental advantage of LayoutLMv2 is that it is very effective in handling noisy and imbalanced datasets. The model was designed to capture all necessary features from the most complex document images that enable the model to yield a high classification rate, making the model highly dependable. Nevertheless, some issues of LayoutLMv2 were mentioned in the study, especially the stability issue that affects its training process. The variation that is observed in the later epochs in terms of accuracy and loss signify that the model could be overfitting or unstable and this can be attributed to many factors such as a high learning rate or complicated data set. These findings suggest that accurate hyperparameter selection and possibly the use of regularization is necessary to stabilize the model.

5.2 VGG16 Performance

VGG16, which is a classic deep learning model of CNN, was used as the benchmark for the present investigation. Nevertheless, the relatively simple compared to LayoutLMv2 network architecture of VGG16 turned out to be quite decent in document classification tasks, reaching the final accuracy of 85% and the final loss of 0.20. The nature where it has 16 layers where each layer is designed similarly enabled the model to extract the features within the document images accurately. However, the changes in accuracy and loss during training implied instability of the model and, therefore, possible weakness of VGG16 to address the peculiarities of document formats compared to LayoutLMv2. When it comes to the more complex layouts and noisy data, VGG16 had some issues to face for the basic document classification tasks. The model's slightly higher loss and less stable accuracy indicate that it might need further fine-tuning or architectural modification to contend with LayoutLMv2 and similar models. However, due to the simplicity and speed of the VGG16, which can be used to detect document layouts and for applications with limited computing power.

5.3 Implications for Document Classification

These implications of this study are therefore of great importance to the field of document classification especially in industries where document categorization is very crucial, for instance, finance, law, and healthcare. Due to its capacity to work with a wide variety of document formats and layouts, LayoutLMv2 has potential in upgrading and enriching the documents sorting and management systems, increasing the classification rate, and optimizing the processes. The improved performance of this system infers that it may be used in organizations where the complexity of the documents is high and precision is of the essence. As for the second, VGG16 may not be the most suitable for large and highly complex documents, yet moderate results obtained by the model suggest that it can still be quite effective in cases where the documents are less intricate in their organization and where the amount of computational power available to process them is limited. This makes VGG16 viable for less resource-endowed organizations or applications where speed of deployment and simplicity trumps the utmost accuracy.

5.4 Limitations of the study

However, this study also found some limitations in the usage of both models even though there were successes noted by the two models. LayoutLMv2 is very efficient but prone to the quality of the OCR outputs. Since the model works with both text and layout information, the presence of errors in the OCR data can greatly reduce the performance of the program. This dependency is a problem when OCR technology fails, for example, for handwritten texts, non-standard fonts, or low-quality images.

However, there is a high computational price of LayoutLMv2, which can be considered as its drawback. The model's architecture is quite complex, and therefore, it needs a great number of resources, such as GPUs or TPUs, much memory, and storage space. This makes it hard to utilize where resources are an issue or where time is of the essence to complete a task or project.

The same applies to VGG16; its drawbacks are linked with its stability and performance when dealing with documents with various formats. The accuracy as well as the loss are oscillating and it can be interpreted as a means that the model may need to be fine-tuned. Moreover, the performance of VGG16 can be less optimal when it comes to the scalability of the system and concerning a broad range of document formats.

Chapter 6

Future Work

6.1 Enhancing Model Stability and Performance

Based on the variations of LayoutLMv2 and VGG16 during training, the research in the future should optimize the model’s stability. For LayoutLMv2, it could be done by trying different learning rates, different levels of regularization, and different dropout techniques to minimize over-fitting and to enhance learning stability. Future work could also focus on identifying the impacts that distinct data augmentation strategies can render to the model to enhance its resistance and ability to generalize.

As for the VGG16, further research could be focused on architectural changes which would allow the model to recognize the more complicated layout of the documents. This can involve such aspects as adding more layers or modules, the primary purpose of which would be capturing the spatial relations within the document images, as well as incorporating such methods as attention mechanisms that would enable the model to pay more attention to specific parts of the document. However, fine-tuning VGG16 on a larger variety of documents and using other optimization techniques might improve its stability and the results overall.

6.2 Enhancing OCR Integration into LayoutLMv2

Since LayoutLMv2 relies on OCR outputs, there may be potential work in improving the interaction of the OCR with the model in the future. This could involve research and the creation of better OCR preprocessing algorithms which would lower the number of mistakes and enhance the standard of the recognized text and the layout data. Moreover, incorporating feedback mechanism where the model itself can correct or enhance the OCR outputs during learning may enhance the classification performance.

Another direction that shows great potential is the integration of the OCR and the document classification problem into a single end-to-end model. Perhaps it is possible to minimize the effect of OCR errors if the OCR module is included as part of the document classification flow. It could also help in minimizing the overall steps that are required for OCR preprocessing and therefore increase efficiency.

6.3 Exploring Scalability Across Diverse Document

For the future work, since the demand for document classification systems increases, especially for multilingual and multi-format documents, further research should be directed at the global application of LayoutLMv2 and VGG16 for documents. This could include feeding the models with higher volumes of data that are made of documents written in various languages, formatted in diverse ways, and having several structures. Further, the studies regarding the improvement of the methods for extending the models to support right-to-left languages, nested tables, and images placed in cells would also be useful.

Thus, the following approaches may help to improve the LayoutLMv2 and VGG16 models for a wider range of applications: Using lesser resources to retrain or fine-tune the model for new document types so that the problem of implementing deep learning models can be addressed. It could include methods like transfer learning where the models are trained in a general set of data and then trained again in a specific kind of document or designing a model that can be trained or updated in some of its components.

6.4 Reducing Computational Costs

That is why the subject of further research should also be the optimization of the costs of the LayoutLMv2 model application. This might include redesigning the model itself to be faster or producing the model’s lite versions that are faster but not necessarily as accurate as the original one. This goal could be achieved with the help of such methods as model quantization, pruning, distillation, where a less complex model is trained to behave in the same way as the more complex model.

Similarly, the study on better hardware and software systems which can cater to the needs of models such as LayoutLMv2 in a more efficient manner may be of immense help when it comes to the overall cost of implementation. This could comprise considering other processing categories like Application-specific Integrated Circuits or AI system processors or even accessing cloud-based processors that can be cheaper sources of high-performance processors.

6.5 Applications Beyond Document Classification

Last, the methodologies and the findings of this research could be applied in other document-related tasks apart from classification. For instance, LayoutLMv2's capability in figuring out the arrangement of a document can be used on tasks like document abstraction, information retrieval or question answering involving visuals. The future work could be the investigation of how the structure and the parameters of the model can be adjusted or whether they can be expanded to cater for these tasks, which can result in the construction of more general and efficient DO systems.

Likewise, VGG16's accuracy in performing image based document tasks could be harnessed to other tasks as well like image segmentation, or object detection on documents. Through such applications, the models derived from this study could extend the spectrum of document processing tools, thereby fertilizing the discriminant applications across the multi-faceted domains.

References

1. Bahdanau, D., Cho, K. and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *arXiv preprint* arXiv:1409.0473.

2. Beltagy, I., Peters, M.E. and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint* arXiv:2004.05150.

3. Breiman, L. (2001). Random forests. *Machine learning*, 45(1), pp. 5-32.

4. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., ... and Amodei, D. (2020). Language models are few-shot learners. *arXiv preprint* arXiv:2005.14165.

5. Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), pp. 273-297.

6. Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1), pp. 21-27.

7. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V. and Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. *arXiv preprint* arXiv:1901.02860.

8. Devlin, J., Chang, M.W., Lee, K. and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint* arXiv:1810.04805.

9. Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), pp. 1735-1780.

10. Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 328-339.

11. Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. Springer.

12. Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint* arXiv:1408.5882.

13. Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, pp. 1097-1105.

14. Li, J., Liang, J., Qiu, S., Liu, X., Zhou, G. and Tang, S. (2021). VisualBERT: A Simple and Performant Baseline for Vision and Language. *arXiv preprint* arXiv:2105.08512.

15. Liu, J., Shah, M., Jiang, Y. and Li, X. (2019). Learning discriminative representations from RNNs. *IEEE Transactions on Neural Networks and Learning Systems*, 30(7), pp. 1928-1940.

16. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint* arXiv:1907.11692.

17. Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

18. Lundberg, S.M. and Lee, S.I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pp. 4768-4777.

19. Xu, Y., Li, M., Cui, L., Wang, S., Liu, L., Wei, F., & Zhou, M. (2020). LayoutLMv2: Multi-modal pre-training for visually rich document understanding. arXiv preprint arXiv:2012.14740. Available at: https://arxiv.org/abs/2012.14740.

20. McCallum, A. and Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization* (Vol. 752, No. 1, pp. 41-48).

21. Mitchell, T.M. (1997). *Machine learning*. McGraw Hill.

22. Quinlan, J.R. (1986). Induction of decision trees. *Machine learning*, 1(1), pp. 81-106.

23. Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, 34(1), pp. 1-47.

24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., ... and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

25. Xu, Y., Li, M., Zhang, L., Du, B., Cui, L., Chang, B. and Wei, F. (2020). LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint* arXiv:2012.14740.

26. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. and Le, Q.V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. *arXiv preprint* arXiv:1906.08237.

27. Yogatama, D. and Smith, N.A. (2014). Making the most of bag of words: Sentence regularization with alternating direction method of multipliers. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)* (pp. 656-664).

28. Zaheer, M., Guruganesh, G., Dubey, K.A., Ainslie, J., Alberti, C., Ontanon, S., ... and Ahmed, A. (2020). Big bird: Transformers for longer sequences. *arXiv preprint* arXiv:2007.14062.

29. Zhang, Y. and Yang, Q. (2021). A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 34(12), pp. 5586-5609.

30. Zhou, J., Ma, Y., Wang, C., You, Q. and Zhao, L. (2020). Hierarchical transformers for long document classification. *arXiv preprint* arXiv:2012.14756.

31. Zhu, J., Ahmed, A., Lu, W. and Zhao, L. (2020). TwinBERT: Distilling Knowledge to Twin-structured BERT Models for Efficient Retrieval. *arXiv preprint* arXiv:2012.14737.

32. Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A. and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE*

international conference on computer vision (pp. 19-27).

33. Zoph, B., Vasudevan, V., Shlens, J. and Le, Q.V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8697-8710).

34. Zuo, W., Luo, W. and He, D. (2019). Attention-based bidirectional LSTM model for indoor user movement recognition using smartphone sensors. *Sensors*, 19(4), p. 79

Appendix

Code

Click Here

Click Here