WORLD COMET RESEARCH GROUP

International Journal of Applied Data Science & Modern Computing
Vol. 4, No. 2, 2021
Doi: 10.67228/30713498/IJADSMC-2021PII3D5V
PP. 01-17

Original Article

Deep Learning Models for Document Classification

Dr. Siti Aisyah Rahman,

Department of Education, Nusantara State University, Kuala Lumpur, Malaysia.

Received: 08-02-2021 Revised: 08-24-2021 Accepted: 09-01-2021 Published: 09-04-2021

ABSTRACT

Document classification is an essential process of natural language processing (NLP) which presupposes assigning textual documents to predefined categories depending on their contents. As the amount of digital text that needs to be classified has grown exponentially through the sources of social media, scholarly repositories, law archives, news portals and enterprise document management systems, effective and correct document classification has become more important. Most of the common machine learning models such as Naïve Bayes, Support Vector Machines, and k-Nearest Neighbors have proven to be of acceptable performance but they heavily depend on manually crafted features and are not as accurate in detecting semantic and contextual information in text. The latest technology in deep learning greatly altered the methods of document classification as it allowed extracting features and learning representations based on the context. Convolutional neural networks (CNNs) models, recurrent neural networks (RNNs), Long Short Memory networks (LSTMs), Gated Recurrent Units (GRUs) and transformer-based have been used to set the state of the art on benchmark datasets. Such models employ dense word encodings, attention, and hierarchical models to represent document-level semantic structures on both local and global levels. In the present paper, the systematic investigation of the document classification frameworks using deep learning models is offered. It analyzes background information, architectural design, learning process and optimization plans. Moreover, it evaluates the advantages and weaknesses of the different deep learning methods in processing long texts, multi-label classification, domain adaptation, and scalability. The socio-economic metrics are also standard performance metrics, and a single approach to the methodology is suggested that incorporates preprocessing, embedding learning, model training, and evaluation using these metrics. The results of the experiment when exploring representative datasets are addressed to emphasize the trends of the comparative performance. The paper will end by presenting some of the current challenges and the direction of future research and stressing the aspects of explainability, efficiency, and domain robustness.

KEYWORDS

Document Classification, Deep Learning, Natural Language Processing, CNN, RNN, Transformer, Text Mining.

WCRG
Global Research
ISSN: 3071-3498
PP: 01-17.

1. INTRODUCTION

1.1. Background

The sudden digitalization of information over the last decades have seen unstructured textual information generated due to varying sources increasing at a rate never seen before namely news media, social networks, scholarly publications, and enterprise repositories. With the continued increase of content and variety of the texts, the notion of hand-classifying documents has grown more and more inappropriate, since it is expensive, not designable in large-scale, and subjective itself. The need to achieve high-speed, uniform and scalable document classification of large document sets, economic human labor and error minimization have thus made automated document classification systems a critical solution. Such systems are important when it comes to many applications such as information retrieval, content recommendation, sentiment analysis, and knowledge management. Early methods of document classification were mostly rule-based, based on manually developed rules and domain-specific lexicons. Though useful in small-scale. Scopes, they could not be subsistence and flexible and, as a result, they were hard to be extended to other areas and changing data distributions. As a response to these weaknesses statistical techniques for machine-learning, including Naïve Bayes, Support Vector Machines, and decision tree-based K estimators were brought into the limelight. These models provided better capabilities of generalization and less dependency on strict rules, but they performed significantly better with feature engineering, such as bag-of-words, TF-IDF, and n-gram features. These characteristics tend to disregard the word order, contextual dependencies as well as the more profound semantic associations with the result of information loss and lack of expressiveness. The key innovation provided by deep learning models is that hierarchical and semantically enriched representations can be automatically learned using only raw text in document classification. Convolutional, recurrent and transformer-based architectures are effective in capturing the localities, sequence of context and global dependency without much feature design art. Large-scale labelled datasets as well as the extensive pretrained word and contextual embeddings and high-performance computer systems have contributed further to the rapid pace of absorbing deep learning methods and made it the current state of the art in contemporary document classification systems.

1.2. Importance of Deep Learning Models

Deep learning models have now taken the center stage of contemporary document classification as they can automatically learn not only intricate patterns and representations but also on textual large-scale data. In comparison with classical machine learning methods, i.e. handcrafted features, shallow representations, in deep learning architectures hierarchical and context-sensitive features are mined out of raw text. This has made it much easier to classify, enhance its stability and flexibility in different fields of use.

ISSN: 3071-3498
PP: 01-17.

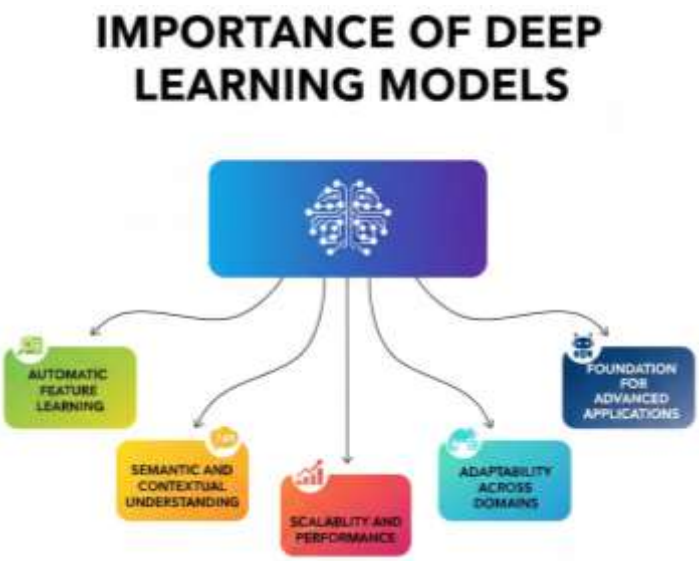

Fig 1 - Importance of Deep Learning Models

1.2.1. Automatic Feature Learning

Deep learning models have the privilege of automatically learning features and this is one of the most significant benefits. Deep neural networks do not rely on manually devised features, like bag-of-words features or n-grams, but instead obtain discriminative features through training. Some of the basic patterns of lexicon are acquired at lower layers and syntactic patterns and semantic ideas are represented at higher layers. The hierarchical representation minimizes the human effort involved in features engineering and also improves in model generalization across datasets.

1.2.2. Semantic and Contextual Understanding

Deep learning models perform well in semantic relationship and contextual dependency between text. Transformer-based and recurrent neural network architectures are based on the idea that such words order, long-range dependencies, and contextual interpretation so a single word could be perceived differently in contexts where it was used. This situational awareness has played a very important role in interpreting subtle language phenomena like ambiguity, sarcasm, and hidden sentiment which is most of the times difficult to the traditional classifier.

1.2.3. Scalability and Performance

Deep learning models are able to scale to large volumes of documents with the existence of large datasets and the development of parallel computing. CNNs and other architectures can be used to perform efficient parallel processing, whereas pretrained transformer models can take use of transfer learning to perform well with limited labelled data. Consequently, the deep learning models are shown to be better than the traditional models on all mainstream benchmarks of document classification.

ISSN: 3071-3498
PP: 01-17.

1.2.4. Adaptability Across Domains

Deep learning models are highly adaptable to new topics and tasks. Language models and pretrained embeddings can be trained to be fined tuned with applications without the large-scale retraining required to build a model starting with a blank-slate. This enables profound learning based classifiers to be presented in thousands of fields, including medical, financial, legal, and social media analytics.

1.2.5. Foundation for Advanced Applications

In addition to classification, deep learning models are used to provide the basis of complex applications of natural language processing, including information extraction, question answering, and text summarization. The representational ability allows them to be smoothly incorporated into downstream functions as they form a foundation block of intelligent text analytics systems.

1.3. Challenges in Document Classification

Although much has been achieved on automated document classification, it still has a number of underlying challenges that further restrain model performance and generalization. The first and the main one is the dimensions of texts and their sparsity especially in the bag-of-words and TF-IDF in conventional representations in a vector space. The vocabulary of textual data can be large and feature groups with tens of thousands or millions of dimensions, the majority of which take the value of zero. This sparsity causes more computational complexity and prevents an effective learning process and renders models more vulnerable to overfitting particularly when the amount of labeled data is thin. The other significant problem is the problem of semantic ambiguity and contextual dependency in natural language. The equivalence of words is possible to different meanings depending on the individual use, words may also have the same idea which has various lexical forms. Conventional classifiers find it difficult to embrace these subtleties where those frequently consider words independent of each other and do not offer an account of the word order. Even sophisticated models need to successfully process implicit meanings, idiomatic phrases and long-range dependencies in an attempt to come up with accurate classification. Very close to this is the problem of variable document length whereby documents may take the form of short messages, to long reports. The ability to design models that could always carefully retrieve relevant information at different lengths without loss of crucial context or creating noise is still not a trivial objective. Also, the class imbalance and multi label classification conditions are major challenges. In practice, real-life examples tend to be skewed with only a small number of examples representing some categories and a large number of examples representing others. This imbalance may bias models towards the majority classes, and worse off, performance on minority categories. The multi-label setting also introduces a certain level of difficulty to the learning process and evaluation of documents that can belong to more than one category. Lastly, the domain shift and vocabulary mismatch is also a persistent challenge when models learned on a specific domain are deployed to a different domain. The variation in writing style, terminology, and article distributions can result into performance

ISSN: 3071-3498
PP: 01-17.

declines as well, which explains why solid and flexible models that can generalize across domains and adapt to the use of language may be required.

2. LITERATURE SURVEY

2.1. Traditional Machine Learning Approaches

The common approach to traditional document classification methods makes use of the use of the vectors space models to store the textual data in numerical form. ST Bag-of-Words (BoW), Term Frequency – Inverse Document Frequency (TF-IDF) and n-gram are common forms of feature representation that transform documents into sparse vectors of high dimension. Naïve Bayes (NB), Support Vector Machines (SVM), Decision Trees, and k-nearest neighbor (k-NN) represent only a few examples of the popular classical machine learning algorithms that have been widely used on these representations. Amongst them, Naïve Bayes was popular by its probabilistic basis and computing ability, whereas SVMs were shown to be strong in high-dimensional spaces with limited training samples. Although their application as baseline classifiers is effective, these methods require high levels of manual feature engineering and domain knowledge. They therefore have trouble in capturing semantic relationships and are affected by sparsity which causes information loss and encounter poor performance when placed upon long documents or text that depends on the situation hence it limits their generalization and scalability.

2.2. Neural Network-Based Models

With the introduction of neural network models, the result of document classification changed drastically since it learned features automatically using raw text. In sharp contrast with other methods, neural models use dense word representations, including Word2Vec and GloVe who encode semantic and syntactic closeness among words in continuous vectors. These representations also enable models to be more generalizable across domains and minimize the use of handcrafted features. The neural architectures brought about a novel nonlinear transformation that enhanced better representation learning and improved classification, especially in complex linguistic scenarios. Consequently, there emerged more sophisticated deep learning methods that are enabled by neural networks that can capture both level of uncertainty and contextual effects in documents.

2.2.1. Convolutional Neural Networks

One of the earliest deep learning models that were successfully applied to a task of classifying a text was Convolutional Neural Networks (CNNs). Here, CNNs are used to compute local characteristics based on the n-gram patterns and based on the sequence of word embeddings. These filters obtain discriminate phrases, and local contextual information that is very informative on classification. The method using CNNs is computationally efficient because of the parallel processing nature and less parameters are required than recurrent architectures. They have been proven to be effective on tasks that involve short or moderately-sized texts, e.g. news categorization and sentiment analysis. Nevertheless, CNNs themselves are already restricted in their ability to learn long-range

ISSN: 3071-3498
PP: 01-17.

interactions and context of a document since their receptivity block is determined by filter size and pooling procedures.

2.2.2. Recurrent Neural Networks

Recurrent Neural Networks (RNNs) introduced sequential modeling features because it works with text by processing each word sequentially, hence maintaining the word sequence and context flow. The types Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) were created to help overcome the challenge of vanishing gradient of the traditional RNNs, making it possible to model long-term dependencies. Such architectures have performed better with respect to syntax and semantic relations among sentences. RNN-based models have proven quite useful in activities that require contextual insight across long sequences. However, since they are sequential, they incur greater computational expenses and require increased time to be trained, in particular when the document is long, and hence are less scalable to large datasets.

2.3. Hierarchical Models

Hierarchical models were suggested to give a clear picture of the natural structure of a document which is naturally established around words, sentences and paragraphs. Hierarchical Attention Networks (HANs) are a remarkable improvement through the use of a two-level architecture: a word-level encoder that builds the representations of sentences and sentence-level encoder that builds the representations of the documents. To pinpoint and highlight the most informative words and sentences that lead to the final classification, attention mechanisms are reinforced at any of the levels. This is a hierarchical design that enhances interpretability, especially of long documents. The added architectural complexity and training procedure in stages however lead to increased computational overhead and necessitate specific hyperparameter optimization.

2.4. Transformer-Based Models

Transformer-based models are defined by the fact that they have revolutionized document classification substituting recurrence, or the convolution with a self-attention mechanism. Each word observes all other words in a document and this modeling of long-range dependencies and contextual interactions is what self-attention ensures. Large-scale unsupervised training without supervision Pretrained language models like BERT and RoBERTa use the roots of large language models to learn deep contextual representations, which can be fine-tuned to a particular classification task. These models always achieve better results than their former counterparts in a number of benchmarks because they have a better contextual knowledge and bi-directional coding. Transformer-based models are resource-intensive models that need massive computational capabilities and memory resources, which makes them challenging to implement in resource-starved models.

ISSN: 3071-3498
PP: 01-17.

2.5. Comparative Summary

Overall, methods of document classification have changed over the years as traditional methods based on features have given way to the use of deep learning models that are built on the concepts of representation learning. CNNs are efficient and highly perform on short text and the RNN and LSTMs model sequential context at the expense of computational complexity. Hierarchical models enhance document level understanding with the use of structural information but have problems in training. Resource-demanding Self-attention and contextual embeddings make transformer-based architectures state-of-the-art in terms of performance. This has raised the trade-off between model complexity, interpretability, computational cost, and classification accuracy, which may stimulate further research into efficient and scalable document classification models.

3. METHODOLOGY

3.1. Data Collection and Preprocessing

Preparation of data is an important step of pipelines that use documents classifier because the quality of the input text directly affects the output. Plain textual information obtained in a variety of sources including news stories, term papers, or other repositories usually include noise, contradiction, and redundant content. Thus, systematic preprocessing strategy is used to normalize text, eliminate redundancy, and improve semantic clarity using the data before inputting it into machine learning or deep learning models.

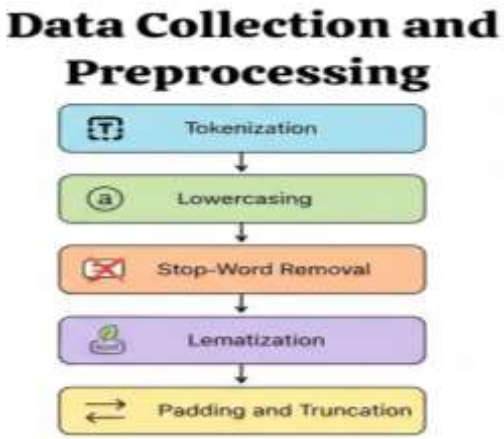

Fig 2 - Data Collection and Preprocessing

3.1.1. Tokenization

The procedure of dividing the unstructured text into small parts called tokens, usually referring to words or subwords, is referred to as tokenization. The step provides the ability of models to manipulate the text in a disciplined manner by converting continuous text streams into discrete units. Traditional machine learning methods typically use word-level tokenization, but modern deep learning and transformer-based systems normally use subword tokenization methods to make use of infrequent words and out-of-vocabulary words. Correct tokenization guarantees consistency in text representation and is used as the backbone to the later operations of preprocessing.

ISSN: 3071-3498
PP: 01-17.

3.1.2. Lowercasing

Lowercasing is the process that involves changing all the characters of the content to lower case to have consistency among tokens. This process is done to decrease the vocabulary size of words like data which could also be represented as data, which further reduces the redundancy of features. Small lettering also especially helps with traditional and neural models, which do not necessarily represent the sensitivity of the case. Nevertheless, it is used wisely in the transformer based models and over normalization could eliminate valuable information in activity where capitalization holds a semantic value.

3.1.3. Stop-Word Removal

Stop-word removal does away with repetitive words that do not carry any semantic meaning of a document like the, is, and and, etc. Stop word elimination can decrease dimensionalities and computation costs particularly in the case of a TF-IDF and other representations based on vectors. The process assists models with concentration on informative terms that are extra discriminating towards classification tasks. However, removing stop-words is commonly not done or done selectively in deep learning models because contextual embeddings are capable of learning to attach significance to them.

3.1.4. Lemmatization

Lemmatization Linguistic normalization Lemmas are the base or canonical form of words. Indicatively, the word running, ran and runs are shrunk to one lemma run. Lemmation Unlike in stemming, the grammatical context of words is taken into account, and more interpretable and significant changes are made. This action alleviates vocabulary sparsity and enhances generalization because the semantically similar words forms are evenly treated during model training.

3.1.5. Padding and Truncation

Padding and truncation will be used, to make the length of all input sequences the same, a shift to the requirement of even-length input sequences in neural networks that will be run in batches. Padding entails the addition of special tokens to short sequences and truncation entails the removal of surplus tokens in the long texts. This standardization allows the efficient training and inference as well as being compatible with model inputs of fixed size. The maximum sequence length is very much required to be selected carefully and it should be an occurrence that is computed with care to balance between computational efficiency and the preservation of information especially in long document classification process.

3.2. Text Representation

Text representation is one of the most important document classification tasks to process unstructured text information into numerical vectors which can be transformed by machine learning and deep learning algorithms. Sparsity Thick representations Sparsity This limitation is addressed within densely-vectorized bag-of-words models to learn semantic and syntactic word-word

ISSN: 3071-3498
PP: 01-17.

relationships. The contextual and static embedding techniques are both taken into consideration in this study to give rich and informative description of documents.

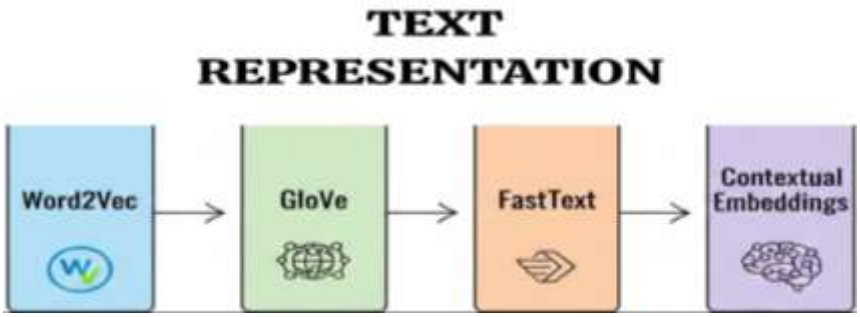

Fig 3 - Text Representation

3.2.1. Word2Vec

Word2Vec is a predictive embedding model, which learns dense vectors of words by reusing their co-occurrence statistics in the context of words in a large corpus. It uses two structures, i.e. Continuous Bag-of-Words (CBOW) and Skip-Gram to model semantic similarity, which scans the words around. Words that are in related contexts are mapped to corresponding points in the embedding space making it possible to do semantic generalization. Word2Vec embeddings are computationally inexpensive and can be utilised in neural text classification models, but they produce one fixed-point vector related to a word, irrespective of the context.

3.2.2. GloVe

Global Vectors of Word Representation (GloVe) is a count-based embedding algorithm that is a model that takes into lexical co-occurrence statistics over the world and neighborhood contexts. GloVe learns linear word-word relationships, including analogical similarities, semantic similarities, and so on, by factorizing a global co-occurrence matrix. These embeddings have shown itself to be quite good at several tasks in natural language processing, and present stable representations that can be used in downstream classification. Like Word2Vec, GloVe generates context-free embeddings, which restricts its capability to capture polysemy and contextual issues.

3.2.3. FastText

FastText builds on classical word embedding models to include subword-level information, i.e. character n-grams. The design enables FastText to come up with meaningful embeddings of rare and out-of-vocabulary words by describing them by smaller linguistic units. Consequently, FastText should be used as an algorithm with whatsoever languages are morphologically rich and with whatsoever text data is noisy. It has the capacity to make use of both semantic and morphological features making it more robust and the classification performance effective than purely word-level embeddings.

ISSN: 3071-3498
PP: 01-17.

3.2.4. Contextual Embeddings

Contextual embeddings are used to overcome these constraints of the static word vectors by creating word vectors, which change actively as the context changes. BERT and ELMo models, and RoBERTa generate various representations of the same word based on its use in a sentence. Such context sensitive representation makes the accurate modeling of polysemy, long-range and accurate semantics possible. Contextual embeddings have greatly enhanced document classification accuracy especially in long and intricate text setting, but at the sacrifice of higher computational and memory loads.

3.3. Model Architecture

The suggested document classification scheme adheres to the generalized deep learning system intended to learn the hierarchical and contextual representations of textual information in a successful manner. The model consists of a set of various interdependent layers, which perform a particular transformation of the input text, with low-level semantic encoding to high-level decision making. This scalable design makes it flexible and adaptable to other deep learning backbones, such as CNNs, RNNs and transformer-based models.

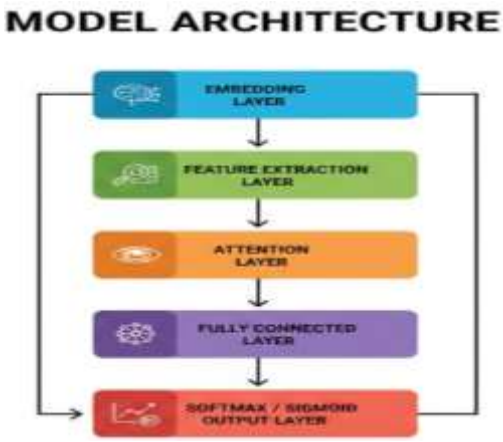

Fig 4 - Model Architecture

3.3.1. Embedding Layer

The embedding layer is the first part of the model and it converts discrete tokens into continuous dense vector representations. Words or subword tokens are mapped to a fixed-dimensional embedding vector that is meant to encode semantic and syntactic data. These embeddings can be initialized with the help of pretrained representations like Word2Vec, GloVe, or FastText, or trained dynamically. The embedding layer minimizes the sparsity of inputs and gives the input the relevant numerical representation that enhances feature learning in the following layers.

3.3.2. Feature Extraction Layer

The task of the feature extraction layer is to extract informative patterns and dependencies of the embedded text representations. This layer can take the form of a convolutional neural network (to discover local n-gram features), a recurrent neural network (to model local sequential dependencies),

ISSN: 3071-3498
PP: 01-17.

or a transformer encoder (to discover the global contextual association using self-attention). This layer is important in identifying the discriminative textual features that are significant in the classification activities by learning their hierarchical features representations.

3.3.3. Attention Layer

The attention layer improves the capacity of the model to pay more attention to the most informative sections of a document, because the salient words or sentences have a greater weight. This process can help the model to focus the light on relevant context information and dispell irrelevant paragraphs. Attention has been noted to be especially useful in long document handling, enhancing interpretability and also enabling global semantic relevance to be taken into consideration by the model without necessarily using sequential processing. Consequently, attention mechanisms have a heavy influence on better classification accuracy and strength.

3.3.4. Fully Connected Layer

The layer of full connectivity combines the obtained features and conducts classification at a high level. It implements linear transformations then nonlinear activation functions to make learned representations directly represent a set of decision space. This layer combines data of the prior layers and allows this model to acquire complex decision boundaries. At this step, regularization methods like dropout are commonly used to avoid overfitting and enhance the performance of generalization.

3.3.5. Sigmoid / Softmax Output Layer

Final classification probabilities are generated in the output layer depending on the task formulation. In multi-class classification, there is a softmax activation function which produces a normalized probability distribution across all possible classes. Conversely, the sigmoid activation is used in binary or multi-label classification tasks whereby each class can independently be estimated in terms of probability. This layer transforms the learned representations of the model into interpretable representations, which can be used to make effective decisions.

3.4. Loss Function and Optimization

The strategy of loss function and optimization is an important part of a deep learning-based document classification model that may be highly effective or not. In classification problems that include multiple classes, categorical cross-entropy loss has been commonly used since it offers an intuitive metric of the difference between the actual distribution of classes and the predicted one produced by the model. In this expression, the loss is calculated to be the negative value of the summations, over all classes, of the true label indicator times the logarithm of the actual predicted probability of the class. Simply put, on each training sample, the loss punishes the model in case it gives a small probability of the correct class, and rewards in case the correct class is given a big probability. This summed up over all classes gives the total loss, and it is important to make sure that the model learns to generate well-calibrated probability distribution. The categorical cross-entropy is especially suited to the use of softmax-based output layers, because it is directly proportional to the

ISSN: 3071-3498
PP: 01-17.

probabilistic interpretations and promotes the separation of classes during its training. In order to reduce the loss-function and optimally estimate model parameters, the gradient-based optimization algorithms are used. Among them, Adam and RMSProp are the popular ones because of their processes of adaptive learning rate. Adam, short for Adaptive Moment Estimation, calculates the learning rate of every parameter by keeping exponentially decadent averages of both the first-order gradients and the second-order squared gradients. The converge more quickly, is more stable, and sturdy to noisy gradient is a benefit of this dual-moment estimation, which renders Adam appropriate to large-scale text classification. RMSProp, however, varies the learning rate according to a moving average of squared gradients, which essentially averts outliers in large parameter updates, resulting in better convergence of deep networks. Both optimizers have lower requirements of learning rates to be manually adjusted and are also successful in non-stationary optimization surfaces. As a result, categorical cross-entropy loss with adaptive methods of optimization guarantee training stability, effective convergence, and higher levels of generalization in models of multi-class document classification.

3.5. Evaluation Metrics

To determine the performance of document classification models, there needs to be working and interpretable metrics that are able to capture the overall performance and the performance of the model at the class level. As classification tasks frequently have a class imbalance and different error cost, more than one evaluation measure is used in order to offer a complete estimate of the model effectiveness.

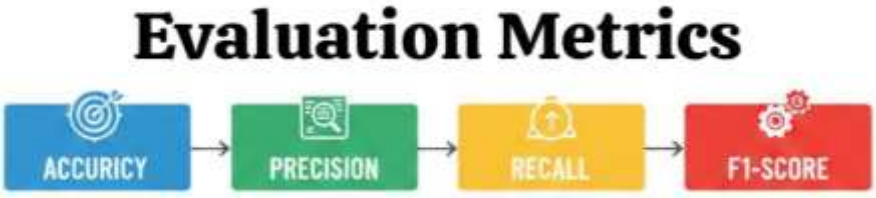

Fig 5 - Evaluation Metrics

3.5.1. Accuracy

The accuracy measures the percentage of well-classified papers among all the papers considered. It gives a simple pointer of general model performance and it comes in handy when there is a symmetric distribution of classes. Accuracy is calculated by finding the number of correct predictions divided by the total amount of predictions. Although this measure is a simple measurement, it can give inaccurate results when there is no balance in the dataset, and a model can become very accurate by doing so by biases towards the majority class.

3.5.2. Precision

Precision will measure the correctness of positive predictions of the model. It is determined as the number of true positive predictions to the summation of number of times that are predicted to be positive. High precision means a small false positive rate and this is considered to be of significance

ISSN: 3071-3498
PP: 01-17.

in studies where false positive is very severe. Precision gives an understanding of how well the model is reliable with its positive predictions and it is the opposite side of accuracy which is concerned with the quality of a prediction but not its quantity.

3.5.3. Recall

Recall, which is also called sensitivity, is the measure of the capability of the model to recognize any other relevant instance of a certain class correctly. It is calculated by the division of the number of true positive predictions with the total number of the actual positive instances. High recall value implies that the model has been able to capture most of the related documents only eliminating false negatives. This measure is essential in a situation whereby false alarms are less expensive than losing key cases.

3.5.4. F1-Score

The F 1 -score is computed as a harmonic mean of precision and recall as it can be used to offer a balanced view of a model performance. The F1-score combines both measures into a single score, which results in a penalty on extreme cases of quantifying precision and recall. It is especially a relevant metric to measure document classification models on unbalanced datasets because it captures the trade-off between the ability to identify relevant documents correctly and the ability to exclude incorrect predictions. Consequently, the F1-score is highly used as a strong and beneficial performance measure in text classification studies.

4. RESULTS AND DISCUSSION

4.1. Experimental Setup

To enable determining the reliability, reproducibility, and comparability with already published studies, the experimental evaluation is done based on the globally accepted benchmark datasets to classify documents. Particularly, the 20 Newsgroups, Reuters-21578, and AG News data sets are used in the experimental work each of which covers different features in terms of length of the documents, variety of topics, and diversity of classes. Multiple datasets allow to provide the overall evaluation of the generalization of the proposed model in a situation of heterogeneous text classification. The Newsgroups 20 Newsgroups data set is a relatively small set of documents amounting to around twenty thousand vehicles that are assigned twenty different levels of thematic topics, including politics, science, religion and technology. The documents are quite different in length and type of writing, and therefore this dataset is especially effective in assessing the capability of a model to retrieve the contextual and semantic information of long and noisy texts. Reuters-21578 data represents newswire stories divided by several topics in the field of finance, economics and trade. Reuters-21578 is a multi-labeled dataset with class imbalance; therefore it is often used to validate the strength of classification models in real and adverse circumstances. By contrast, the AG News data is a large, high-balanced corpus of text consisting of news articles, and it consists of four high-level categories that are world, sports, business, and science/technology. Its texts are relatively short and organised thus making it the best in testing the model performance of the short text

ISSN: 3071-3498
PP: 01-17.

segment in classifying documents. All datasets undergo techniques of standard preprocessing such as tokenizing, normalizing, and standardizing sequence length. To make the evaluation fair, the datasets are split into training, validation, and test subdivisions based on the commonly used splits. The hyperparameters that are optimized using a validation set are modeled and the final performance is reported on the held-out test set. Through these benchmark datasets, standardized experimental protocols will enable the experimental set-up to be an objective and rigorous framework of testing the effectiveness of the suggested document classification strategy.

4.2. Performance Comparison

In this section, the various document classification models are analyzed in comparison with regard to their accuracy and F1-score. The analysis demonstrates the gradual increase in the performance of the models as they become more sophisticated in the models of deep learning and transformers.

Table 1: Performance Comparison

Model	Accuracy (%)	F1-Score (%)
SVM (TF-IDF)	84.2	83
CNN	88.7	88
LSTM	89.5	89
HAN	91.3	91
Transformer	94.6	94

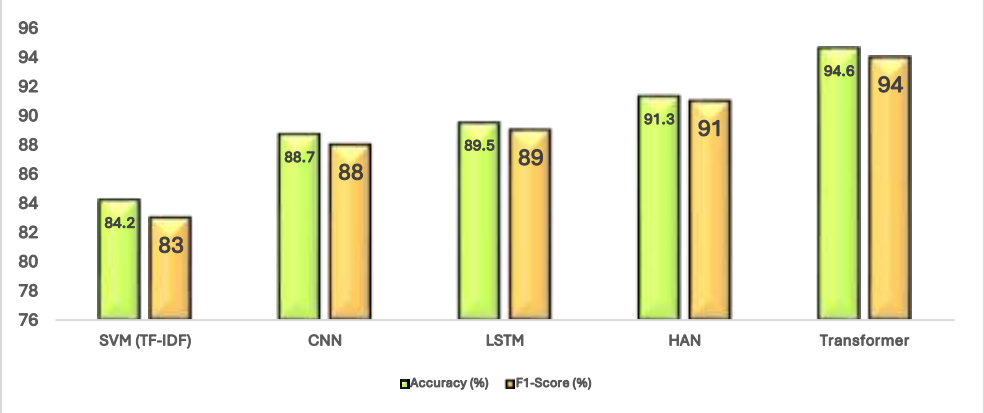

Fig 6 - Performance Comparison

4.2.1. SVM (TF-IDF)

The TF-IDF features with Support Vector Machine (SVM) model provide an accuracy score of 84.2 and F1-score of 83.0, which is a good starting point when it comes to classifying documents. Those findings indicate that the margin-based classifiers are effective in high-dimensional feature spaces. Nonetheless, the use of sparse representations and hand constructed features constrain the capacity of the model to model semantic connectivity and contextual information causing relatively poorer performance than neural models.

ISSN: 3071-3498
PP: 01-17.

4.2.2. Convolutional Neural Network (CNN)

CNN-based classifier achieves the accuracy of 88.7% and F1-score of 88.0 which is a significant increase in comparison with traditional methods. CNNs are useful in learning local n-gram characteristics by applying convolutional embeddings in texts, thereby extracting discriminative linguistic patterns. Their parallel computing facilitates the effective education of them by large datasets. However, additional performance improvements are hampered by the fact that the possibility to model long-range dependencies is low.

4.2.3. Long Short-Term Memory (LSTM)

With LSTM model, 89.5 percent of the accuracy and 89.0 percent of the F1-score are attained, which is higher than CNNs, as sequential dependencies and contextual flow are modeled within documents. Gated architecture of LSTM makes it store long-term information which is more useful in capturing sentence level semantics. Regardless of such benefits, the sequential character of LSTMs leads to a rise in the complexity of computations and the time required to train.

4.2.4. Hierarchical Attention Network (HAN)

Accuracy (91.3) and F1-score (91.0) of Hierarchical Attention Network proves that it is beneficial to explicitly model document structure. In encoding words into sentences and documents into documents, HAN is capable of well capturing hierarchical information and sets prominental textual elements with the assistance of attention process. This hierarchical model results in better classification accuracy particularly with long documents, but at the cost of architectural and training complexity.

4.2.5. Transformer

The transformer-based model is the most with high performance with accuracy of 94.6 and F1-score of 94.0. Such high performance is due to the self attention mechanism that makes it possible to model long range dependencies and deep interactions of context across the whole document. Representation quality can also be enhanced with pretrained contextual embeddings, which ensures that transformers can be better than previous architectures at all times. Nonetheless, such advantages are at the expense of higher computational and memory demands.

4.3. Discussion

The experimental findings clearly show that, the transformer based models are on average better at document classification in comparison with other traditional machine learning models and previous deep learning models. This high level of performance is due to their advantage of self-attention which allows a complete model of contextual dependencies of complete-document scope. In comparison with CNNs and RNNs, transformers do not use any fixed-size receptive fields or sequential processing, but instead each token can attend any other token at the same time. Such awareness of contexts across the globe is especially useful in capturing the long-range semantic relations, uncovering ambiguity, and modeling complex linguistic patterns, a fact that relates to the

ISSN: 3071-3498
PP: 01-17.

tremendous gains in the accuracy and the F1-score. Transformer-based models have significant practical challenges, even despite these benefits. Their accuracy is achieved at a very high cost of high computational and memory demands, particularly at the pretraining and fine-tuning phases. Use of quadratic complexity of self-attention in relation to the input length further escalates resource consumption to longer documents thus making deployment hard on real-time or resource starved settings. Also, transformers can be more susceptible to large labeled datasets and hyperparameters fine-tuning, so their application to specific fields can be constrained by this factor. On the contrary, the convolutional neural networks are still an appealing approach when it comes to real-time and large-scale use because of their low computing power and simultaneous processing. CNNs are also efficient in capturing local patterns and salient n-gram information, which allows performing inferences quickly and consume comparatively low resources. They might not bring the contextual performance and efficiency of the transformers, but their ability to balance between performance and efficiency makes them applicable in other operations like internet filtering of content and news classifications. On the whole, the discussion informs about a trade-off between performance and the computational cost with a main focus on the need to select a proper model architecture depending on the application needs, data specifications, and resources at hand.

5. CONCLUSION

In this paper, a thorough study of document classification methods has been provided, with the emphasis put on the development of the traditional methods of machine learning to up-to-date deep learning and transformer-based frameworks. It was evidenced through comprehensive literature review and experimental examination that deep learning models are far superior to classical methods since they automatically acquire, through textual information, rich semantic and contextual representations. Deep architectures like CNNs, RNNs, hierarchical models, and transformers are effective representations of complex patterns in linguistics, long-range dependencies, and hierarchical organizations of documents, unlike traditional models that make heavy use of handcrafted features and minimal representations. Transformer based models had the best classification performance, which reflects their potential of modeling the context of the world and subtle semantic dynamics of an entire document. Although these significant developments, there are still several challenges that inhibit the extensive use of the deep learning models in document classification. Model interpretability is one of these issues, as deep architectures are commonly black boxes, and are not easily understandable or explainable on policy classification decisions. This lack of transparency has challenges in the areas of critical uses, e.g. legal analysis, and healthcare documentation as well as financial reporting which require explainability. Computational efficiency presents another challenge especially in models that are transformer based since they consume a large amount of memory, processing power, and even energy. These limitations do not allow them to be used in real-time systems and resource-constrained environments. Moreover, the area of adaptability of domains is not completely closed, as when the models are applied to new areas or changing data distributions, the model performance tends to decline. Future studies must overcome these shortcomings and solve the challenges of explainable document classification, which

ISSN: 3071-3498
PP: 01-17.

incorporates attention visualization, feature attribution, and explainable neural parts of the problem to increase further transparency and user confidence. Another aspect of how to reduce computational overhead and stay competitive while providing competitive performance is the development of lightweight transformer models, including parameter-efficient models and knowledge-distilled models. Also, it is essential to expand document classification systems to enable cross-linguistic and multilinguistic classification to ensure the increasing amount of universal and heterogeneous textual information. Lastly, continual and federated learning methods provide practical solutions to the dynamism of data streams and transferring models to a decentralized setup without violating data privacy. All these research directions lead to more effective, understandable and flexible document classification systems that could respond to the real-life needs.

REFERENCES

[1] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," Proc. 10th European Conf. Machine Learning (ECML), pp. 137–142, 1998.
[2] A. McCallum and K. Nigam, "A comparison of event models for Naïve Bayes text classification," AAAI Workshop on Learning for Text Categorization, pp. 41–48, 1998.
[3] Y. Yang and X. Liu, "A re-examination of text categorization methods," Proc. 22nd Int. ACM SIGIR Conf., pp. 42–49, 1999.
[4] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," Communications of the ACM, vol. 18, no. 11, pp. 613–620, 1975.
[5] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," Proc. NAACL-HLT, pp. 103–112, 2015.
[6] Y. Kim, "Convolutional neural networks for sentence classification," Proc. EMNLP, pp. 1746–1751, 2014.
[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," Proc. ICLR, 2013.
[8] J. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[9] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," Proc. EMNLP, pp. 1724–1734, 2014.
[10] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," Proc. NAACL-HLT, pp. 1480–1489, 2016.
[11] A. Vaswani et al., "Attention is all you need," Proc. Advances in Neural Information Processing Systems (NeurIPS), pp. 5998–6008, 2017.
[12] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," Proc. NAACL-HLT, pp. 4171–4186, 2019.
[13] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," arXiv preprint arXiv:1907.11692, 2019.
[14] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," Proc. ICML, pp. 1188–1196, 2014.
[15] S. Minaee et al., "Deep learning-based text classification: A comprehensive review," IEEE Access, vol. 8, pp. 12956–13013, 2020.