symmetry

MDPI

Article

Contextual Embeddings-Based Web Page Categorization Using the Fine-Tune BERT Model

Amit Kumar Nandanwar * **and Jaytrilok Choudhary**

Computer Science & Engineering Department, Maulana Azad National Institute of Technology, Bhopal 462003, India
* Correspondence: amitdataset@gmail.com; Tel.: +91-9827730050

Abstract: The World Wide Web has revolutionized the way we live, causing the number of web pages to increase exponentially. The web provides access to a tremendous amount of information, so it is difficult for internet users to locate accurate and useful information on the web. In order to categorize pages accurately based on the queries of users, methods of categorizing web pages need to be developed. The text content of web pages plays a significant role in the categorization of web pages. If a word's position is altered within a sentence, causing a change in the interpretation of that sentence, this phenomenon is called polysemy. In web page categorization, the polysemy property causes ambiguity and is referred to as the polysemy problem. This paper proposes a fine-tuned model to solve the polysemy problem, using contextual embeddings created by the symmetry multi-head encoder layer of the Bidirectional Encoder Representations from Transformers (BERT). The effectiveness of the proposed model was evaluated by using the benchmark datasets for web page categorization, i.e., WebKB and DMOZ. Furthermore, the experiment series also fine-tuned the proposed model's hyperparameters to achieve 96.00% and 84.00% F1-Scores, respectively, demonstrating the proposed model's importance compared to baseline approaches based on machine learning and deep learning.

Keywords: BERT; BiLSTM; contextual embedding; deep learning; DMOZ; WebKB; web page categorization

check for updates

Citation: Nandanwar, A.K.; Choudhary, J. Contextual Embeddings-Based Web Page Categorization Using the Fine-Tune BERT Model. *Symmetry* **2023**, *15*, 395. https://doi.org/10.3390/sym15020395

Academic Editor: Dumitru Baleanu

Received: 10 January 2023
Revised: 24 January 2023
Accepted: 30 January 2023
Published: 2 February 2023

1. Introduction

Digital information such as text, images, audio, and videos on web pages has rapidly increased over the World Wide Web (WWW). Search engines are used to find accurate web pages according to the user's query becomes challenging. For precise and rapid search engines to perform this task, effective web page categorization techniques are required. Web pages are semi-structured data in a hypertext markup language (HTML) format that can belong to one or more categories containing various elements such as text, images, tables, audio, and video. Web page categorization is a process of putting web pages into their appropriate class [1]. A class can be considered any category of information, such as news, business, sports, entertainment, society, etc. Web page categorization is the state-of-the-art for topic-specific link analysis, focused crawling, contextual advertising, helping question-answering systems, information retrieval, phishing website detection, and search engines [2–4]. Many researchers categorize web pages based on feature vectors produced by traditional machine-learning methods such as term frequency–inverse document frequency (TF-IDF), n-gram, and bag of words (BOW) [5]. Contextual and semantic features extracted from the web pages play a significant role in deciding the actual category of it [6].

Based on the above studies, the problem identified as follows: "Web pages contain various information, and categorizing them is a complex problem". Some web page categorization methods exploit text features with keywords only without considering the context of web pages [7,8]. Deep-learning-based web page categorization is a hot and new research area for efficiently categorizing web pages with improved performance [9–11]. Typically, a fixed vector represents a word without regard for its context in web page categorization. The meaning of the words interpreted from the sentence on the web page is called context. Practically, each word might communicate a varied meaning depending on the context. Search engines cannot find accurate web pages due to varying contextual interpretations, leading to performance degradation.

Natural language processing (NLP) has progressed recently, outperforming all other statistical machine-learning techniques. Along with deep-learning approaches based on convolution neural networks (CNNs), recurrent neural networks (RNNs), and transformers, it provides the most accurate results [12–14]. Web page categorization approaches such as GloVe, word2vec, long short-term memory (LSTM), and bidirectional long short-term memory (BiLSTM) have been used by past researchers by considering the context of their various elements [15,16]. GloVe and word2vec produce the context-free embedding of the contents of web pages [17–19]. In contrast, many feature vectors of words using deeply left to right and right to left with different contexts are produced by BERT [15,20,21].

To categorize web pages based on contextual embedding produced by the BERT model, which is genuinely contextual, was a significant motivation for the present work. BiLSTM is not genuinely bidirectional since the model learns from left to right and right to left separately and then concatenates the context. These shortcomings have led us to move toward a transformer architecture, which addresses some of these issues [22]. BERT-based contextual embedding performed well in many NLP applications, such as text classification and sentiment analysis.

The main objective of the proposed work was to develop an efficient model to categorize web pages based on contextual features produced by BERT. In this paper, we propose a fine-tuned web page categorization model based on BERT contextual embedding with a SoftMax classifier. To enhance the performance of the proposed model, we fine-tuned the max_seq_length hyperparameter.

The main contributions of this paper are as follows:

- We proposed and designed a new model by using BERT for web page Categorization based on contextual features of web pages. As compared to deep-learning models such as LSTM and BiLSTM, the proposed model gained a better performance via fine-tuning.
- The proposed model has a powerful generalization capability and can be widely used in specific domain tasks. A general text corpus trained the BERT pretraining model, and then the model was fine-tuned on a domain-specific corpus (WebKB and DMOZ), effectively improving the model's performance.
- The proposed model's performance is compared with the existing state-of-the-art web page categorization models to demonstrate the generalization potential.

The following is an outline of the remainder of this paper: Related studies that have performed web page categorization are briefly introduced in Section 2. The proposed model is thoroughly described in Section 3. Data description, performance matrices, and hyperparameter setting are discussed in Section 4. The comparative performance analysis and experimental results are discussed in Section 5. Finally, Section 6 provides a conclusion and proposes future research directions.

2. Related Work

The web's popularity has increased recently due to the rapid developments of computer and network technology, such as 4G and 5G. As a result, web page categorization has become an increasingly important issue in recent years. Researchers are putting rigorous efforts into proposing different techniques to handle web page categorization issues and challenges. Earlier researchers developed machine-learning models based on supervised-learning approaches to categorize web pages [1]. Supervised-learning algorithms mainly focused on feature-selection and feature-extraction methods to categorize web pages. Earlier researchers used BOW and TF-IDF techniques to categorize web pages with a traditional machine-learning algorithm [7,8]. In this section, we discuss various research work, providing state-of-the-art proposed work. A complete study of related works is divided into four subsections: feature extraction from the web pages, dimensionality reduction, semantic relationships among the features, and contextual feature-based categorization.

2.1. Feature Extraction and Selection

Lipras et al. [8] proposed a web page categorization model based on a random forest classifier applied to categorize news articles into four categories. These categories are Business–Finance, Lifestyle–Leisure, Science–Technology, and Sports. The frequency of unigram, bigram, trigrams, and four grams feature vectors were utilized to categorize news article web pages. Li et al. [7] applied Naïve Bayesian Classifier to categorize web pages of entity similarity networks based on Wikipedia. Term frequency feature vector obtained with the help of the title and main text of the web pages. Recently Mulahuwaish et al. [23] improved web page categorization by using document frequency-based feature vectors with a Decision Tree (DT), support vector machine (SVM), and K-Nearest Neighbor (KNN).

The most prominent conclusion of these research articles is that BOW and TF-IDF techniques suffer from the dimensionality curse and sparsity problem because of the many features in web pages. Thus, meta-heuristic approaches are utilized by researchers to categorize web pages.

2.2. Dimensionality Reduction of Features

Furthermore, Tian et al. [24] extended research insights by presenting a web page classification model that used the information gain technique to reduce feature dimensionality and improve accuracy. After that, Selamat et al. [25] utilized a principal component analysis (PCA) for dimensionality reduction to improve web page classification accuracy with the feature vectors formed by the TF-IDF and neural network classifier. A simplified swarm optimization (SSO) meta-heuristic approach was proposed by Lee et al. [26] and used to reduce features formed by tagged terms. N. Bacanin et al. [27] presented a novel approach to spam detection based on the combination of machine-learning models and an enhanced sine cosine swarm intelligence meta-heuristic approach. Genetic algorithms, also used by Ozal et al. [28] to categorize web pages, achieved a better classification accuracy than KNN and the naïve Bayesian classifier. Another article proposed an improved web page classification model, using the ant colony algorithm to optimize the feature subset search space [29]. After thoroughly examining previous studies, we found that there was a lack of semantic relationships identified among the text of web pages.

2.3. Semantic Relationships among the Features

At present, many word-embedding methods, such as Word2vec [18], Doc2vec [23], and Glove [17], have emerged as a set of effective representation models for text classification, sentiment analysis, spam classification, and other NLP tasks [30]. In these pretrained models, words are depicted as dense vectors in low-dimensional vector spaces so as to construct a continuous feature representation for texts. However, word embedding has been proven to have a good performance in regard to capturing the semantic information of text units. By incorporating glove-word embeddings into our previous research, we improved web page categorization efficiency through stacked BiLSTM methods [9].

FastText pretrained word embeddings were used by Endalie et al. [13] to capture semantic information from web page text efficiently. It takes subword information by incorporating the character n-gram into the skip-gram model. The deep-learning CNN model's embedding layer extracts higher-level features and categorizes news web pages into six categories. Compared to traditional machine-learning approaches, such as random forests, multilayer perceptrons, support vector machines, and gradient boosting, it achieved a higher F1-Score. Furthermore, Yu [31] used deep-learning models such as CNN and RNN to categorize web pages more efficiently than traditional machine-learning models. This article shows models based on deep learning that extract local and global features from web pages.

The following are some limitations of traditional context-free word-embedding methods, such as word2vec, fast text, and GloVe. In order to generate weight vectors from networks with a narrow depth, it is assumed that each word in the input sequence has a stable meaning. As a result, the polysemy of input tokens and shallow representations of weight vectors are limited. This word-embedding method has proven to be an effective one, but it does not take into account the relationships between multiple words, nor how their placement within a sentence affects the meaning.

2.4. Contextual Features of Web Pages

Consequently, researchers have used context-based word embeddings to improve word embeddings in recent years. Models of natural language are pretrained by using extensive networks and large numbers of unlabeled data. Fine-tuning in downstream tasks has made significant progress in some NLP tasks, including natural language inference, text classification, and textual entailment. The language modeling transformers are deep networks trained on a large text corpus, producing highly accurate contextual text representations and resolving the ambiguity problem in the web page text.

Devlin et al. introduced BERT as the current state-of-the-art embedding model [32,33]. BERT has been adapted in different ways to address problems such as high memory consumption and slow training speeds (i.e., Roberta [34], ALBERT [35], DistilBERT [36], etc.). This paper does not review these studies since it focuses solely on the use of BERT for web page categorization.

An ensemble approach is presented by Gupta and Bhatia [11] for the multiclass categorization of web pages. To categorize web pages into specified categories and to learn the contextual representation of the input, the authors use pretrained BERT. After this representation is created, it is then used as an input for the Deep Residual Inception Model Network.

To extract contextual features from the text content of web pages, Artene et al. [20] used a pretrained multilingual BERT to classify a web page. A convolution filter was applied on these feature embeddings to categorize multi-label and multi-language web pages.

Based on the work of BERT in NLP applications, we propose a model to categorize the web pages of publicly available WebKB and DMOZ datasets. By fine-tuning BERT, we can improve the performance over current state-of-the-art deep-learning and machine-learning models.

3. Methodology

To improve the F1-Score of web page categorization, we developed a novel BERT-based model where BERT is utilized to extract contextual features from web pages. The web page category is predicted by using an output classifier designed for learning by high-level contextual features. The proposed model comprises four parts: data preprocessing, input layer, BERT Encoder layer, and output layer. The architecture of this model is shown in Figure 1.

3.1. Data Preprocessing

Data preparation is the first step in natural language processing and is required before using machine-learning and deep-learning techniques. In this phase, we used a common preprocessing approach that can integrate with various NLP tasks, using NLTK (Natural Language Toolkit) [37]. The web page's contents are organized into HTML, usually semi-structured files containing tags. Text inside the body tags of the web pages is extracted to perform the following preprocessing steps.

- Stop word removal: The primary objective is to reach valid words only by eliminating special characters and words without meaning from input text. Therefore, all punctuation marks, spaces, numbers, and symbols are eliminated from extracted text from the web pages.

- Lowercase: As a result of the stop word removal process, all web page text is converted to lowercase letters based on the language's alphabet. It is an appropriate form for input to the BERT uncased model, which consumes lowercase letters.
- Stemming: At this stage, lemmatization is applied to obtain the simplest form of the word in terms of word roots by removing suffixes. After the suffixes are discarded, tokens are formed as a result of stemming.

Figure 1 illustrates tokens grouped by sentences. For one sentence, tokens are numbered Tok1 to TokN; for another, tokens are numbered Tok1 to TokM.

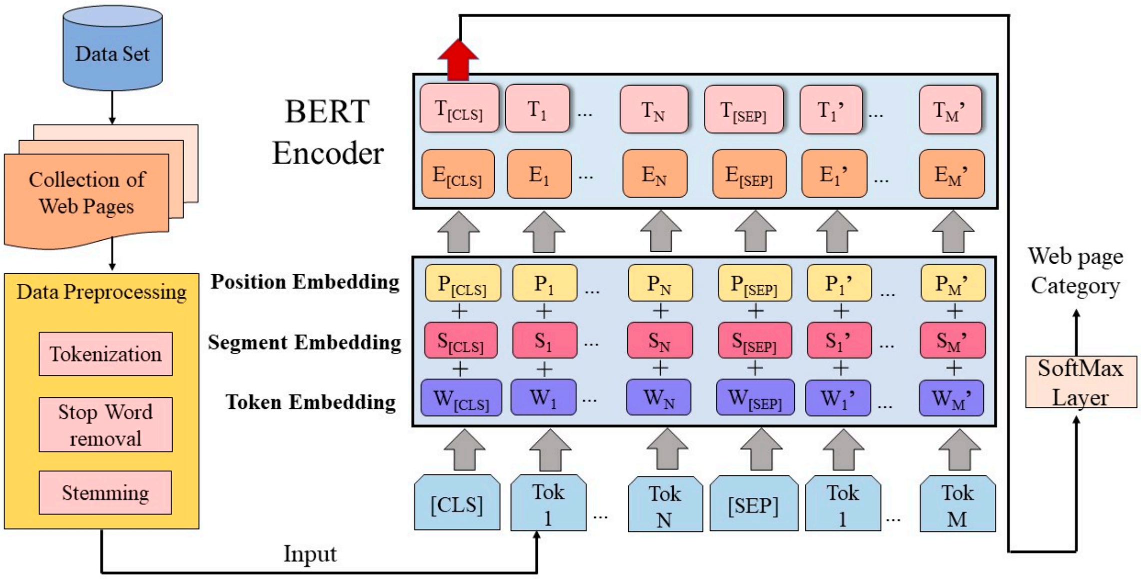

Figure 1. Proposed framework of web page categorization. The red color arrows denotes the final output of the BERT model passed to SoftMax Layer.

3.2. Input Layer

After the data preprocessing, web page text contents are tokenized before being used by an input layer of BERT to build an input sequence for the model. To perform tokenization, an input function named BERT Tokenizer converts tokens into embeddings, a feature vector that extracts the meaning of tokens. Similar tokens are determined based on the closeness of their feature vectors; thus, the model performs efficiently in the numeric form and understands semantic and contextual information.

BERT utilizes a 30,000-token vocabulary to segment the input sequence, using WordPiece embeddings. In this method, a word is fragmented into several tokens, where the first token is the word root, and ## is added before all other tokens of the input word. For example, the word "gaming" is tokenized by using WordPiece embedding as two distinctive tokens, "game" and "##ing", to cover a broad range of Out-of-Vocabulary (OOV) words [38].

The input embedding (E_N) in BERT is the sum of three different embeddings: token embeddings (W_N), segmentation embeddings (S_N), and position embeddings (P_N) for each token. With token embedding, words are transformed into a vector representation of a fixed dimension, which is a vocabulary identifier for each token. Segment embeddings with a shape of (1, n, 768) serve as vector representations to assist BERT in distinguishing between paired input sequences. In addition, transformer positional embeddings indicate the position of each word in the sequence. The input representation for BERT's encoder layer consists of a composite embedding resulting from linear summing of token embedding, segment embedding, and positional embedding, represented as follows:

$$E_N = W_N + S_N + P_N \tag{1}$$

As a result of the summing of these representations, a single representation with shape (1, n, 768) is produced, which is passed to the encoder layer of BERT. In the input representation layer, the BERT model automatically adds the symbols [CLS] and [SEP] at the beginning and end of the sentence to indicate each sentence's beginning and end.

3.3. BERT Encoder Layer

One of the most advanced technologies in the field of natural language processing is the BERT model. In this approach, deep bidirectional representations are pretrained from unlabeled text incorporating both directions of context, and then all parameters of downstream tasks are fine-tuned.

It is a symmetry of a multilayer bidirectional transformer encoder consisting of transformer blocks, a multi-head attention mechanism, and a feed-forward fully connected layer shown in Figure 2.

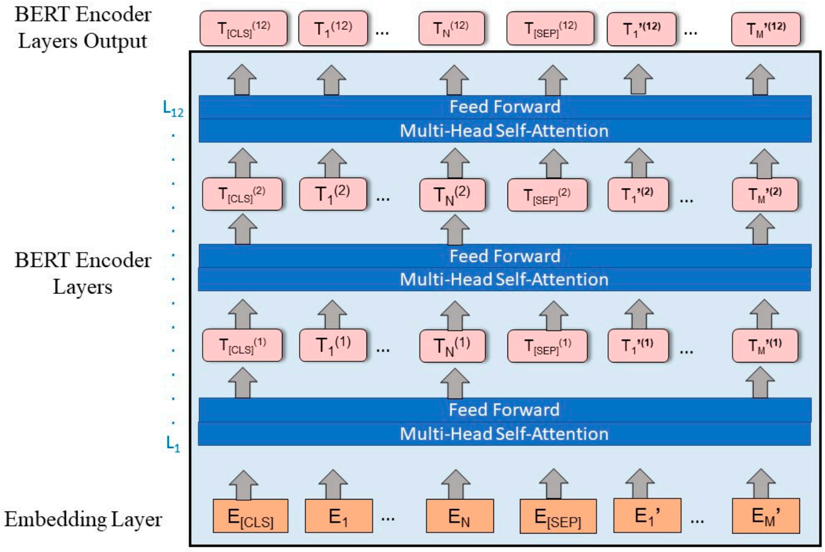
BERT Encoder Layers Output
BERT Encoder Layers
Embedding Layer
L_{12}
L_1
Feed Forward
Multi-Head Self-Attention
$T_{[CLS]}^{(12)}$ $T_1^{(12)}$ ... $T_N^{(12)}$ $T_{[SEP]}^{(12)}$ $T_1'^{(12)}$ ... $T_M'^{(12)}$
$T_{[CLS]}^{(2)}$ $T_1^{(2)}$ ... $T_N^{(2)}$ $T_{[SEP]}^{(2)}$ $T_1'^{(2)}$ ... $T_M'^{(2)}$
$T_{[CLS]}^{(1)}$ $T_1^{(1)}$ ... $T_N^{(1)}$ $T_{[SEP]}^{(1)}$ $T_1'^{(1)}$ ... $T_M'^{(1)}$
$E_{[CLS]}$ E_1 ... E_N $E_{[SEP]}$ E_1' ... E_M'

Figure 2. BERT encoder layer.

The symmetric transformer learns contextual relationships among the words and consists of an encoder that is responsible for reading text input. Polysemes have different meanings in different contexts, and traditional machine-learning methods cannot resolve this problem. Since BERT represents words based on their context, the vector representations of polysemes should vary based on their meaning. The transformer encoder differs from directional models in that it reads all of the words simultaneously, thus allowing it to be non-directional.

In a bidirectional BERT architecture, the context of the word is derived from its surrounding words, resulting in dynamic word vectors. One word can generate distinct vector representations in different contexts, and this significantly improves the expressive ability of word vectors. It not only avoids the possible ambiguity caused by word segmentation but also solves the problem of polysemy. Therefore, the proposed model extracts contextual features from web page text, thereby improving the accuracy of web page categorization.

The proposed model uses the BERT encoder layer to obtain the contextual feature representation of the web page's text. As shown in Figure 2, L denotes the number of layers (transformer blocks), H is the hidden size, and A the number of self-attention heads, where L = 12, H = 768, and A = 12. E_1, E_2, . . . ; E_N represent the input words; and the corresponding vectors' representations, T_1, T_2, . . . , and T_N, are generated after multilayer bidirectional transformer training.

The input vectors E_1, E_2, . . . and E_N are then transformed into queries, Q_m; keys, K_m; and values, V_m, where m ∈ {1 . . . M} represents the *m*th attention head. An M number of parallel attention functions are applied to produce an M number of output states, termed as

T_1, T_2, . . . , and T_M. The transformer encoder layer adopts the Multi-Head Self-Attention mechanism, which is computed as follows:

$$A_m = \text{SoftMax}\ ((Q_m \times K_m{}^{T})/\sqrt{d_k}) \tag{2}$$

$$T_m = A_m \times V_m \tag{3}$$

The attention distribution for the *m*th head is A_m, and $\sqrt{}$dk is a scaling factor. A final output is obtained by concatenating each Ti head:

$$T_i = T_{i1} \ldots \ldots T_{im} \tag{4}$$

Multi-Head Self-Attention can capture the contextual information in the web page's text by considering multiple combinations of target words and other words. The BERT model produces a vectorized representation of the entire sentence as the leftmost [CLS] vector. According to the proposed categorization model, BERT encoders produce the leftmost [CLS] vector that is passed to the output layer.

3.4. Output Layer

The output layer on the top of the BERT encoder used a SoftMax classifier. The BERT encoder output is a sequence of hidden state vectors. The proposed model only uses the final hidden state vector corresponding to [CLS] token representation to predict the probability of each web page. SoftMax uses an exponential activation function to map the input score to the probability that one belongs to the output class category.

Let α be the set of all trainable parameters for the proposed model, and the output layer produces the vector $T_{[CLS]}$ into the probability distribution over all web page categories y = {y_1,y_2,y_3, . . . y_n}, where n = number of categories of web pages, represented as Equation (5):

$$P(y_i/T_{[CLS],}\ \alpha) = \exp\ (P(y_i/T_{[CLS],}\ \alpha))/\ \sum_{i=1}^{n} \exp\ (P(y_i/T_{[CLS],}\ \alpha)) \tag{5}$$

In Equation (6), we take the category with the largest value as the predicted result, since t represents the actual category of web page x:

$$y_x = \text{argmax}\ (P(y_i/T_{[CLS]}, \alpha)) \tag{6}$$

The dropout is adopted as a regularization strategy to avoid overfitting by taking the value of 0.1. The proposed model used the default Adam optimizer with $beta_1$ = 0.9 and $beta_2$ = 0.999. The categorical cross-entropy loss function calculates the loss by computing the following Equation (7):

$$CEL(t,\ y_x) = -\sum_{i=1}^{N} t\ log(y_x) \tag{7}$$

where N is the number of web pages, CEL (*t*, y_x) is the categorical cross-entropy loss, y_{xi} is the predicted output, and t_i is the actual output for the *i*th web page.

4. Experimental Setup

A description of the datasets, performance metrics, and fine-tuning analysis used in the proposed model is presented in this section.

4.1. Datasets

This section describes the summary and statistics of standard web page categorization datasets, such as WebKB [39] and DMOZ [40], used further with the proposed model.

4.1.1. WebKB Dataset

The WebKB dataset consists of web pages collected from computer science departments at the universities of Cornell, Texas, Washington, and Wisconsin, as well as other universities. The web pages are organized into seven categories, including course (930 pages), department (182 pages), faculty (1124 pages), project (504 pages), staff (137 pages), student (1641 pages), and other (3764 pages)". The proposed methodology considers Web pages from four categories, namely "student, faculty, course, and project, for train/test splitting into four categories. The following Table 1 shows the number of train and test web pages in WebKB dataset.

Table 1. The number of trains and tests web pages in WebKB dataset.

Class Label	No. of Web Pages	Train Web Pages	Test Web Pages
Course	930	651	279
Faculty	1124	787	337
Project	504	353	151
Student	1641	1148	493

4.1.2. DMOZ Dataset

DMOZ is a vast benchmark dataset for web page categorization. In the proposed methodology, we used 13 top classes for categorizing web pages into different categories: Business, Society, Science, Recreation, Shopping, Games, Arts, Business, Computers, and Health. The following Table 2 shows the number of train and test web pages in the DMOZ dataset.

Table 2. The number of train and test web pages in the DMOZ dataset.

Class Label	No. of Web Pages	Train Web Pages	Test Web Pages
Arts	193,914	135,740	58,174
Business	204,910	143,437	61,473
Computers	93,204	65,243	27,961
Games	36,454	25,518	10,936
Health	50,388	35,271	15,117
Home	22,241	15,569	6672
News	7488	5242	2246
Recreation	83,618	58,532	25,086
Reference	47,428	33,199	14,229
Science	96,766	67,736	29,030
Shopping	78,184	54,729	23,455
Society	195,880	137,116	58,764
Sports	85,376	59,763	25,613

4.2. Performance Metrics

The purpose of this section is to evaluate the performance of the proposed model in detail, adding a more thorough analysis of the results. The categorization of web pages is a categorization problem. Among the most commonly used evaluations for categorization are accuracy, recall, precision, F1-Score [41], and the confusion matrix [42,43]. This is to ensure that our explanation is clear, as our problem is a multiclass problem, where every web page categorizes exactly one category.

- Confusion matrix: A confusion matrix is used to evaluate the performance of a categorization model. It is represented in matrix form. The confusion matrix compares True Values and predicted values. An N × N matrix represents the confusion matrix, where N is the number of classes. For 2 classes, we obtain a 2 × 2 confusion matrix. TP, FP, TN, and FN are its constituent elements shown in Table 3.

Table 3. Confusion matrix for the binary classification.

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

- Accuracy: Accuracy is a measure of the proportion of correctly classified web pages to the total number of web pages. In general, a classifier with higher accuracy is more accurate.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{8}$$

- Recall: The classifier's sensitivity represented by recall is the ratio of correctly classified web pages to the actual number of web pages. High recall means fewer false negatives.

$$\text{Recall (R)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

- Precision: Precision refers to the proportion of web pages correctly classified to the number of web pages classified.

$$\text{Precision (P)} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{10}$$

- F1-Score: To achieve a balance between precision rate and recall rate, the F1-Score is most commonly used as the evaluation index for categorization tasks. The best value of the F1-Score is 1, and the worst value is 0, which is the harmonic mean of precision and recall.

$$\text{F1} - \text{Score} = \frac{2\text{PR}}{\text{P} + \text{R}} \tag{11}$$

4.3. Fine-Tuning Analysis of the Proposed Model

An integral part of any deep-learning solution is the selection of hyperparameters. In deep-learning algorithms, specific hyperparameters are often defined to control different aspects, such as memory usage and execution time. A hyperparameter is a variable set before an algorithm is applied to a context-specific dataset. The most accurate numbers will differ, as each task and dataset is context dependent. It is possible to select and optimize hyperparameters in two different ways: manually and automatically. Technically, both methods are suitable but represent a trade-off between a deep understanding of the model in order to make manual selections and the high computational load associated with automatic selection procedures. We used a manual approach to select hyperparameter values to achieve accurate web page categorization. Table 4 lists the values of hyperparameters.

Table 4. Values of hyperparameters used during experiments.

Hyperparameter	**Values**
Batch size	32
Learning rate	1×10^{-5}
Epochs	3

To fine-tune the proposed model in the experiments, we observed that most hyperparameters are similar to those used in pretraining, except for batch size, learning rate, number of training epochs, and max_seq_length. Based on the experiments, we can conclude that the max_seq_length hyperparameter significantly influences the performance of the proposed model. Table 5 illustrates how the proposed model is fine-tuned by using max_seq_length hyperparameter scheduling to achieve a better performance on both datasets.

Table 5. Result analysis of the proposed model on WebKB and DMOZ datasets with the different sequence lengths.

Dataset	Max_Seq_Length	Precision	Recall	F1-Score	Accuracy
WebKB	25	0.9300	0.9299	0.9300	0.9261
WebKB	50	0.9500	0.9501	0.9500	0.9484
WebKB	**75**	0.9600	0.9589	**0.9600**	**0.9563**
WebKB	100	0.9600	0.9600	0.9600	0.9501
DMOZ	20	0.8300	0.8200	0.8250	0.8429
DMOZ	**35**	0.8400	0.8350	**0.8400**	**0.8559**
DMOZ	55	0.8401	0.8335	0.8400	0.8539

We evaluated the performance of the proposed model on WebKB and DMOZ datasets by various values of hyperparameter max_seq_length. In WebKB and DMOZ datasets, the average length of the text is 156 and 15.03, respectively. The hyperparameter max_seq_length = 75 on the WebKB dataset, and max_seq_length = 35 on the DMOZ dataset yields the best results in terms of F1-Score and accuracy. Moreover, the values of the dataset, max_seq_length, F1-Score, and accuracy are shown in bold in Table 5.

Considering that the maximum sequence length of the BERT model is equal to 512 preserves all tokens in the sequences without losing any information. In this way, the model is able to better capture the meaning of the sequences and learn more task-specific knowledge, which leads to improved results. It is important to note that when the sentence length is smaller, the model must reject some tokens, and this tends to reduce the model's performance.

BERT models with a greater maximum length provide more aggregated representations and better performance, especially when the training data are insufficient. However, it is important to note that it is accompanied by the cost of rapidly increasing the trainable parameters and training costs. In situations where a sufficient number of supervised training data are available, a longer maximum length might not always guarantee a better performance after exceeding a specific threshold value.

5. Result and Discussion

A performance analysis of the proposed model is presented in this section, using the WebKB and DMOZ datasets. Several methods are compared with the proposed model, including machine-learning classifiers, deep-learning methods, and recent articles.

5.1. Performance of the Proposed Model on the WebKB Dataset

WebKB datasets were used to evaluate the performance of the proposed model. We computed each category's performance metrics, namely the precision, recall, F1-Score, accuracy, and the normalized confusion matrix. The normalized confusion matrix for the proposed model is shown in Figure 3, where the values are normalized by the number of web pages in each category. True positive (TP) value of the Course category is 0.99, indicating that course web pages are categorized most of the time correctly.

Table 6 shows performance metrics for the dataset mentioned in Section 4. The average accuracy and F1-Score for the proposed model on the WebKB test data are 95.63% and 96.00%, respectively.

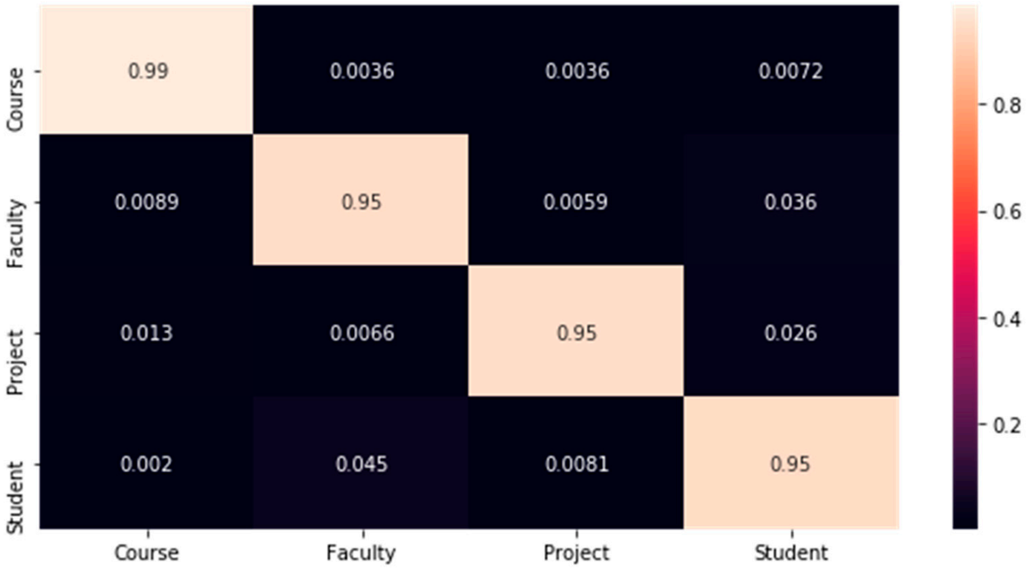

Figure 3. Normalized confusion matrix on WebKB dataset.

Table 6. Precision, recall, F1-Score, and accuracy of the proposed model on the WebKB dataset.

Category	Precision	Recall	F1-Score	Average F1-Score (%)	Average Accuracy (%)
Course	00.98	00.99	00.98	96.00%	95.63%
Faculty	00.93	00.95	00.94		
Project	00.95	00.95	00.95		
Student	00.96	00.95	00.95		

5.2. Performance of the Proposed Model on the DMOZ Dataset

The performance of the proposed web page categorization model was evaluated on the DMOZ dataset. We computed the confusion matrix of thirteen categories of the DMOZ dataset to evaluate performance matrices' precision, recall, and F1-Score. Figure 4 shows that the normalized confusion matrix gives the highest true positive value of the Sports category and the lowest true positive value of the News category. Table 7 shows the performance metric mentioned in Section 4 on the DMOZ dataset and achieved accuracy and F1-Score of 85.59% and 84.00%, respectively.

Table 7. Precision, recall, F1-Score, and accuracy of the proposed model on the DMOZ dataset.

Category	Precision	Recall	F1-Score	Average F1-Score (%)	Average Accuracy (%)
Arts	00.88	00.90	00.89	84.00%	85.59%
Business	00.88	00.87	00.88		
Computers	00.85	00.87	00.86		
Games	00.87	00.85	00.86		
Health	00.84	00.86	00.85		
Home	00.84	00.80	00.82		
News	00.80	00.73	00.76		
Recreation	00.84	00.82	00.83		
Reference	00.72	00.73	00.73		
Science	00.79	00.80	00.81		
Shopping	00.81	00.82	00.82		
Society	00.87	00.86	00.86		
Sports	00.92	00.92	00.92		

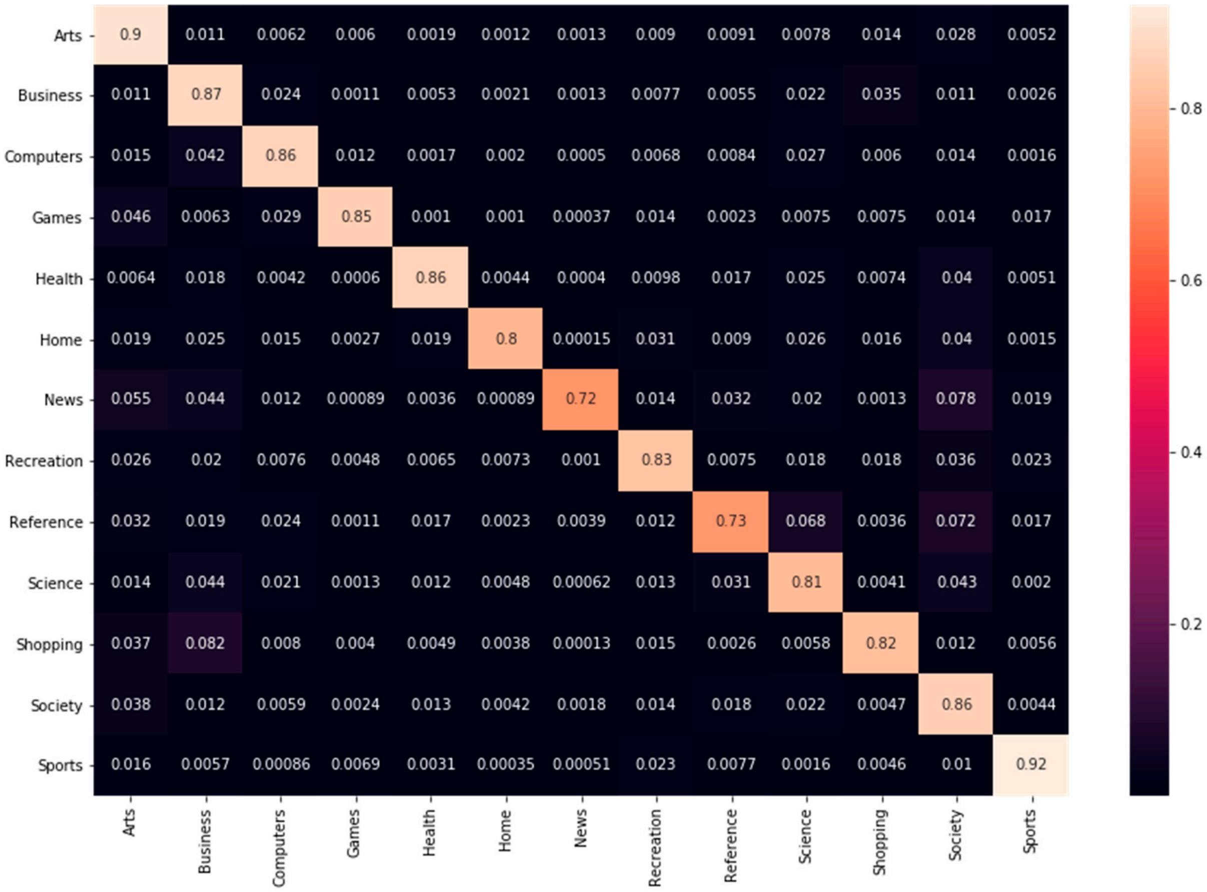

Figure 4. Normalized confusion matrix on DMOZ dataset.

5.3. Performance Comparison with the Baseline Models

The performance evaluation was established with two benchmark datasets for web page categorization to evaluate the proposed model's effectiveness. Figures 5 and 6 depict the comparison of the proposed model with the state-of-the-art methods in terms of precision, recall, F1-Score, and accuracy. As of this writing, the state-of-the-art techniques include deep-learning and machine-learning models based on supervised learning for categorizing web pages as follows:

- Support vector machines: SVM is a discriminative classification method commonly recognized as more accurate. The SVM classification method is based on the computational learning theory's Structural Risk Minimization principle [11,13,44]. This principle aims to find a hypothesis to guarantee the lowest actual error. The optimal hyperplane created by SVM categorizes the web pages. In this paper, the SVM classifier classifies the web pages based on the input feature vector formed by the TF-IDF method with the redial bias function kernel and achieved an F1-Score of 77.00% and 69.00% on the WebKB and DMOZ datasets, respectively.
- The naive Bayes: It is helpful in text classification based on the Bayes theorem [45]. One of the advantages of naive Bayes classifiers is that it requires a small number of training data. It uses a probabilistic approach to categorization. The naive Bayes assumes that the input features are independent of one another. The category with the highest probability then determines the label for the sample. The MNB classifier utilized the feature vector formed by the TF-IDF method to categorize web pages and achieved an F1-Score of 75.00% and 67.00% on the WebKB and DMOZ datasets, respectively.

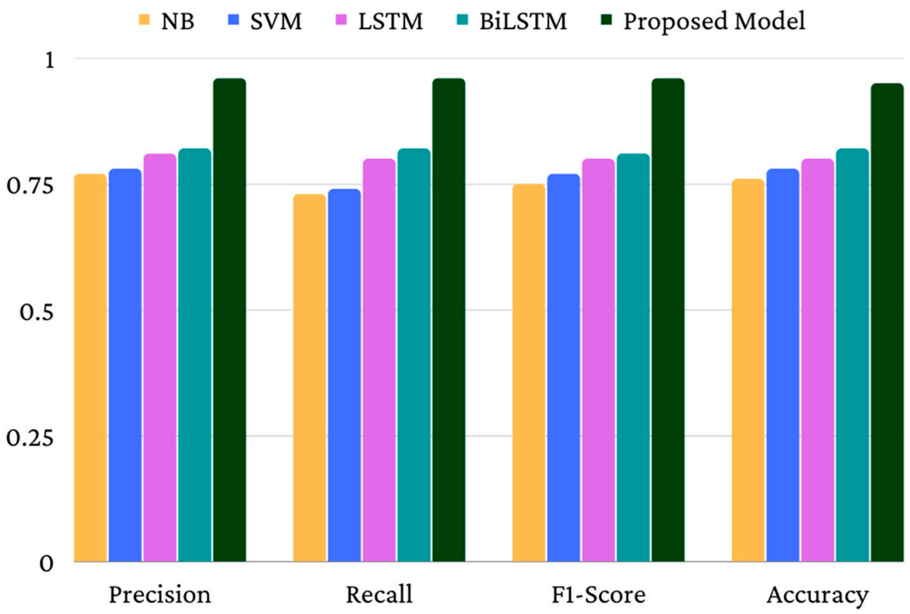

Figure 5. Comparative performance analysis of the proposed model with baseline models on WebKB.

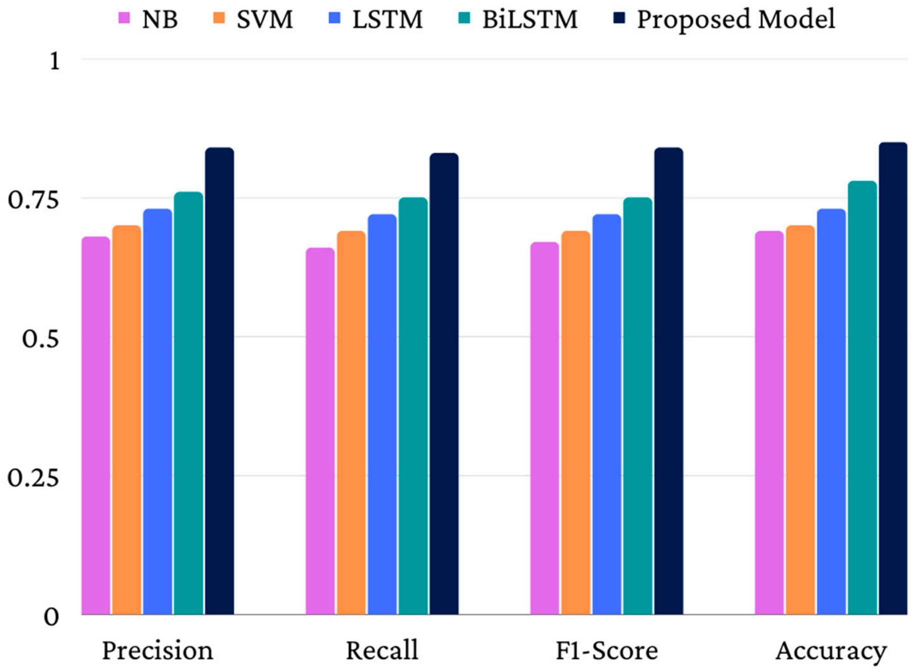

Figure 6. Comparative performance analysis of the proposed model with baseline models on DMOZ.

Statistical machine learning has been replaced by deep learning based on text categorization models in current state-of-the-art approaches [33]. The feature vector form by the BOW and TF-IDF methods lacks contextual features and data sparsity problems. The data sparsity problem was resolved by the word-embedding technique mentioned in the proceeding section.

- LSTM: Long short-term memory (LSTM) is an exceptional category of Recurrent Neural Networks due to its ability to avoid vanishing gradient problems [6,46]. LSTM can use long memory in the hidden layer as the input for the activation function. These have been mainly used to develop an end-to-end deep neural network to extract contextual features from raw text. In this implementation, the word-embedding layer has been utilized to form feature vectors which are passed as input to the LSTM. It categorizes the input category as output with the softMax output layer. The LSTM model captures contextual information from data in the left to the right direction and achieved an F1-Score of 80.00% and 72.00% on WebKB and DMOZ datasets, respectively. Thus, the model's performance improved with respect to the SVM and NB machine-learning models.

- BiLSTM: The application of BiLSTM [16,47] to categorize web pages is to increase its performance. This is because the BiLSTM model contains two independent LSTM models to capture the contextual information the model will learn from left to right and right to left separately. Then it joins the two together. Therefore, we used the BiLSTM model to categorize web pages to achieve an F1-Score of 81.00% and 75.00% on WebKB and DMOZ datasets, respectively.

The proposed model was implemented by using the BERT model and the SoftMax output layer on the WebKB and DMOZ datasets for categorizing web pages. To improve the performance of the proposed model, fine-tune it with the hyperparameter max_seq_length.

According to Figures 5 and 6, the proposed model achieved a maximum accuracy of 95.63% and 85.59% compared to other machine-learning and deep-learning models on WebKB and DMOZ, respectively.

5.4. Performance Comparison with Recently Published Methods

Tables 8 and 9 show that the proposed model's performance has been compared to the recently published article on precision, recall, F1-Score, and accuracy.

Table 8. Comparative performance comparison on WebKB dataset.

Author (s)	Method	Precision	Recall	Accuracy	F1-Score
Bhalla et al. [48]	SVM	0.9233	0.8233	-	0.8633
Gupta et al. [11]	BERT, Convolution Layer, Inception Layer	0.3900	0.2200	0.7900	0.1300
Nandanwar et al. [9]	GloVe, Stacked BiLSTM	0.8401	0.8239	0.8532	0.8303
Proposed model	BERT, SoftMax	0.9600	0.9600	0.9563	0.9600

Table 9. Comparative performance comparison on the DMOZ dataset.

Author(s)	Method	Precision	Recall	Accuracy	F1-Score
Gupta et al. [11]	BERT, Convolution Layer, Inception Layer	0.6150	0.3290	0.6600	0.6350
Nandanwar et al. [9]	GloVe, Stacked BiLSTM	0.7834	0.7708	0.8023	0.7749
Proposed model	BERT, SoftMax	0.8400	0.8300	0.8559	0.8400

In Table 8, the contextual feature-based proposed model shows a 96.00% F1-Score, which is superior to the recently published articles dealing with the WebKB dataset. These articles were chosen for comparison because they also work with related deep-learning and machine learning methods, but our fine-tuned proposed model shows an improved F1-Score. Specifically, the methods of Bhalla et al. [48], showing 86.33% and using SVM; Gupta et al. [11], offering 13.00%, using BERT with Inception; and our previous article [9], showing 83.03%, was based on GloVe being underperformed compared to the proposed model.

In Table 9, experimental results are compiled by using another benchmarked DMOZ dataset and performance comparisons with previous research works. These results are presented to demonstrate the proposed model's generalization capability. While working with the DMOZ dataset, the proposed method is still the best performer in terms of the F1- Score. In comparison with the technique of Gupta et al. [11], showing 63.50%, and the model in our previous article [9], showing 77.49%, the proposed method showed an 84.00% score and was proved to be superior.

6. Conclusions

With advances in Internet technology, users are unable to find accurate information on the World Wide Web. The accuracy of existing web page categorizations needs to be further improved. The improvement in the web page categorization model affects the performance of information retrieval systems, such as search engines. Search engines use the categorized web pages to sort and find more relevant results on user queries. Toward the achievement of this objective, a new web page categorization model was proposed in this paper. BERT with a SoftMax layer was utilized to categorize the web pages as the categorization model. BERT learns contextualized word representations from many unlabeled text datasets due to its complex structure and excellent nonlinear representation learning capability. The proposed model effectively improves performance by memorizing crucial information and finding patterns from unlabeled text data. The experiment was performed on two publicly available web page datasets, WebKB and DMOZ, and achieved improved performance over existing methods, deep-learning models, and other machine-learning classifiers. Implementing the proposed model on the WebKB dataset gave an accuracy and F1-Score of 95.63% and 96.00%, respectively. For the generalization, the proposed model also applied to the DMOZ dataset gave an accuracy and F1-Score of 85.59% and 84.00%, respectively.

As the contents of web pages have diverse information, different feature-combination methods, including contextual and semantic features, can be applied to improve categorization performance in the future. Additionally, proposed model tests with more datasets related to the business and medical field may further verify the effectiveness and generalization of the model.

Author Contributions: Conceptualization, A.K.N.; methodology, A.K.N.; software, A.K.N.; validation, A.K.N. and J.C.; formal analysis, A.K.N.; investigation, A.K.N.; writing—original draft preparation, A.K.N.; writing—review and editing, A.K.N. and J.C.; visualization, A.K.N. and J.C.; supervision, J.C.; project administration, J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hashemi, M. Web Page Classification: A Survey of Perspectives, Gaps, and Future Directions. *Multimed. Tools Appl.* **2020**, *79*, 11921–11945. [CrossRef]
2. Qi, X.; Davison, B.D. Web Page Classification. *ACM Comput. Surv.* **2009**, *41*, 1–31. [CrossRef]
3. Yu, S.; Su, J.; Luo, D. Improving BERT-Based Text Classification with Auxiliary Sentence and Domain Knowledge. *IEEE Access* **2019**, *7*, 176600–176612. [CrossRef]
4. Tang, L.; Mahmoud, Q.H. A Survey of Machine Learning-Based Solutions for Phishing Website Detection. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 672–694. [CrossRef]
5. Sánchez, J.; Perronnin, F.; Mensink, T.; Verbeek, J. Image Classification with the Fisher Vector: Theory and Practice. *Int. J. Comput. Vis.* **2013**, *105*, 222–245. [CrossRef]
6. Liu, G.; Guo, J. Bidirectional LSTM with Attention Mechanism and Convolutional Layer for Text Classification. *Neurocomputing* **2019**, *337*, 325–338. [CrossRef]
7. Li, H.; Xu, Z.; Li, T.; Sun, G.; Raymond Choo, K.K. An Optimized Approach for Massive Web Page Classification Using Entity Similarity Based on Semantic Network. *Future Gener. Comput. Syst.* **2017**, *76*, 510–518. [CrossRef]
8. Liparas, D.; HaCohen-Kerner, Y.; Moumtzidou, A.; Vrochidis, S.; Kompatsiaris, I. News Articles Classification Using Random Forests and Weighted Multimodal Features. In *Multidisciplinary Information Retrieval*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2014; Volume 8849, pp. 63–75. [CrossRef]
9. Nandanwar, A.K.; Choudhary, J. Semantic Features with Contextual Knowledge-Based Web Page Categorization Using the GloVe Model and Stacked BiLSTM. *Symmetry* **2021**, *13*, 1772. [CrossRef]
10. Nandanwar, A.K.; Choudhary, J. Web Page Categorization Based on Images as Multimedia Visual Feature Using Deep Convolution Neural Network. *Int. J. Emerg. Technol.* **2020**, *11*, 619–625.
11. Gupta, A.; Bhatia, R. Ensemble Approach for Web Page Classification. *Multimed. Tools Appl.* **2021**, *80*, 25219–25240. [CrossRef]

12. Shivakumara, P.; Tang, D.; Asadzadehkaljahi, M.; Lu, T.; Pal, U.; Hossein Anisi, M. CNN-RNN Based Method for License Plate Recognition. *CAAI Trans. Intell. Technol.* **2018**, *3*, 169–175. [CrossRef]
13. Endalie, D.; Haile, G. Automated Amharic News Categorization Using Deep Learning Models. *Comput. Intell. Neurosci.* **2021**, *2021*, 3774607. [CrossRef] [PubMed]
14. Kaliyar, R.K.; Goswami, A.; Narang, P.; Sinha, S. FNDNet–A Deep Convolutional Neural Network for Fake News Detection. *Cogn. Syst. Res.* **2020**, *61*, 32–44. [CrossRef]
15. Geetha, M.P.; Karthika Renuka, D. Improving the Performance of Aspect Based Sentiment Analysis Using Fine-Tuned Bert Base Uncased Model. *Int. J. Intell. Netw.* **2021**, *2*, 64–69. [CrossRef]
16. Hameed, Z.; Garcia-Zapirain, B. Sentiment Classification Using a Single-Layered BiLSTM Model. *IEEE Access* **2020**, *8*, 73992–74001. [CrossRef]
17. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 26–28 October 2014; pp. 1532–1543.
18. Zhao, W.; Zhu, L.; Wang, M.; Zhang, X.; Zhang, J. WTL-CNN: A News Text Classification Method of Convolutional Neural Network Based on Weighted Word Embedding. *Connect. Sci.* **2022**, *34*, 2291–2312. [CrossRef]
19. Badri, N.; Kboubi, F.; Chaibi, A.H. Combining FastText and Glove Word Embedding for Offensive and Hate Speech Text Detection. *Procedia Comput. Sci.* **2022**, *207*, 769–778. [CrossRef]
20. Artene, C.G.; Tibeica, M.N.; Leon, F. Using BERT for Multi-Label Multi-Language Web Page Classification. In Proceedings of the 2021 IEEE 17th International Conference on Intelligent Computer Communication and Processing ICCP 2021, Cluj-Napoca, Romania, 28–30 October 2021; pp. 307–312. [CrossRef]
21. Rai, N.; Kumar, D.; Kaushik, N.; Raj, C.; Ali, A. Fake News Classification Using Transformer Based Enhanced LSTM and BERT. *Int. J. Cogn. Comput. Eng.* **2022**, *3*, 98–105. [CrossRef]
22. Selvakumar, B.; Lakshmanan, B. Sentimental Analysis on User's Reviews Using BERT. *Mater. Today Proc.* **2022**, *62*, 4931–4935. [CrossRef]
23. Mulahuwaish, A.; Gyorick, K.; Ghafoor, K.Z.; Maghdid, H.S.; Rawat, D.B. Efficient Classification Model of Web News Documents Using Machine Learning Algorithms for Accurate Information. *Comput. Secur.* **2020**, *98*, 102006. [CrossRef]
24. Tian, L.; Zheng, D.; Zhu, C. Image Classification Based on the Combination of Text Features and Visual Features. *Int. J. Intell. Syst.* **2013**, *28*, 242–256. [CrossRef]
25. Selamat, A. Web Page Feature Selection and Classification Using Neural Networks. *Inf. Sci.* **2004**, *158*, 69–88. [CrossRef]
26. Lee, J.H.; Yeh, W.C.; Chuang, M.C. Web Page Classification Based on a Simplified Swarm Optimization. *Appl. Math. Comput.* **2015**, *270*, 13–24. [CrossRef]
27. Bacanin, N.; Zivkovic, M.; Stoean, C.; Antonijevic, M.; Janicijevic, S.; Sarac, M.; Strumberger, I. Application of Natural Language Processing and Machine Learning Boosted with Swarm Intelligence for Spam Email Filtering. *Mathematics* **2022**, *10*, 4173. [CrossRef]
28. Özel, S.A. A Web Page Classification System Based on a Genetic Algorithm Using Tagged-Terms as Features. *Expert Syst. Appl.* **2011**, *38*, 3407–3415. [CrossRef]
29. Saraç, E.; Özel, S.A. An Ant Colony Optimization Based Feature Selection for Web Page Classification. *Sci. World J.* **2014**, *2014*, 649260. [CrossRef]
30. Guo, Y.; Mustafaoglu, Z.; Koundal, D. Spam Detection Using Bidirectional Transformers and Machine Learning Classifier Algorithms. *J. Comput. Cogn. Eng.* **2022**. [CrossRef]
31. Yu, Y. Web Page Classification Algorithm Based on Deep Learning. *Comput. Intell. Neurosci.* **2022**, *2022*, 9534918. [CrossRef]
32. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL HLT 2019-2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 4171–4186.
33. Minaee, S.; Kalchbrenner, N.; Cambria, E.; Nikzad, N.; Chenaghlu, M.; Gao, J. Deep Learning Based Text Classification: A Comprehensive Review. *ACM Comput. Surv.* **2020**, *54*, 1–40. [CrossRef]
34. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
35. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations. *arXiv* **2019**, arXiv:1909.11942.
36. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *arXiv* **2019**, arXiv:1910.01108.
37. Li, C.; Liu, K. Smart Search Engine: A Design and Test of Intelligent Search of News with Classification. Bachelor's Thesis, Dalarna University, Falun, Sweden, 2021.
38. Subba, B.; Kumari, S. A Heterogeneous Stacking Ensemble Based Sentiment Analysis Framework Using Multiple Word Embeddings. *Comput. Intell.* **2022**, *38*, 530–559. [CrossRef]
39. McCallum. The 4 Universities Data Set. Available online: http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/ (accessed on 12 July 2021).
40. DMOZ-The Directory of the Web. Available online: https://www.dmoz-odp.org/ (accessed on 16 August 2021).

41. Vishwakarma, G.; Thakur, G.S. Hybrid System for MPAA Ratings of Movie Clips Using Support Vector Machine. In *Advances in Intelligent Systems and Computing*; Springer: Singapore, 2019; Volume 817, pp. 563–575.
42. Banerjee, I.; Ling, Y.; Chen, M.C.; Hasan, S.A.; Langlotz, C.P.; Moradzadeh, N.; Chapman, B.; Amrhein, T.; Mong, D.; Rubin, D.L.; et al. Comparative Effectiveness of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) Architectures for Radiology Text Report Classification. *Artif. Intell. Med.* **2019**, *97*, 79–88. [CrossRef]
43. Solanki, S.; Dehalwar, V.; Choudhary, J. Deep Learning for Spectrum Sensing in Cognitive Radio. *Symmetry* **2021**, *13*, 147. [CrossRef]
44. Vishwakarma, G.; Thakur, G.S. Comparative Performance Analysis of Combined Svm-Pca for Content-Based Video Classification by Utilizing Inception V3. *Int. J. Emerg. Technol.* **2019**, *10*, 397–403.
45. El Hindi, K.M.; Aljulaidan, R.R.; AlSalman, H. Lazy Fine-Tuning Algorithms for Naïve Bayesian Text Classification. *Appl. Soft Comput. J.* **2020**, *96*, 106652. [CrossRef]
46. Brahma, B.; Wadhvani, R. Solar Irradiance Forecasting Based on Deep Learning Methodologies and Multi-Site Data. *Symmetry* **2020**, *12*, 1830. [CrossRef]
47. El-Alami, F.Z.; Ouatik El Alaoui, S.; En Nahnahi, N. Contextual Semantic Embeddings Based on Fine-Tuned AraBERT Model for Arabic Text Multi-Class Categorization. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 8422–8428. [CrossRef]
48. Bhalla, V.K.; Kumar, N. An Efficient Scheme for Automatic Web Pages Categorization Using the Support Vector Machine. *New Rev. Hypermedia Multimed.* **2016**, *22*, 223–242. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.