Adv Syst Sci Appl 2024; **02**; 133-165
Published online at https://ijassa.ipu.ru.

Text Classification Technologies in Document Categorization Systems. A Survey

Alla Kravets, Dmitriy Semenochkin

Volgograd State Technical University, Volgograd, Russia

Abstract: This paper presents a literature review from 2013 to 2022 on technologies and datasets used in the field of text classification. The review covered 110 sources from 5 scientific databases, the main criterion for inclusion was the presence of an experimental part involving a classifier or other technologies related to the classification process. The study reviewed the classification process and highlighted three main stages of text classification: data preparation, classifier training, and evaluation of results. Using Kitchenham's Systematic Literature Review methodology, scholarly articles dealing with text classification problems were collected and analyzed. A sample of 243 articles was obtained, and after screening, a resulting sample of 110 articles was obtained. Guided by the two research questions posed, this sample was analyzed and the results of the analysis were presented in graphical format. For each of the identified stages of classification, the frequency of use of the main technologies used in a particular stage was analyzed. Each technology was reviewed within its respective source. In addition, considerable attention was given to analyzing the different datasets used for text classification, with a particular focus on the less frequently used ones. An analysis of the frequency of use of datasets concluded that researchers often use proven and popular datasets to demonstrate the effectiveness of their method. Datasets are less frequently used to solve localized text classification problems. One notable trend identified in the analysis is the increasing prevalence of deep learning technologies in text classification. These technologies, including neural networks, recurrent neural networks (RNNs), convolutional neural networks (CNNs), transformers, and attention mechanisms, have gained considerable popularity among researchers. This study provides valuable insights into the evolution of text classification by shedding light on a variety of technologies, approaches, and datasets used by researchers. As text classification continues to evolve and diversify, this review can be a valuable resource for scholars and practitioners in the field, providing.

Keywords: Text Classification, Text Classifiers, Deep Learning, Classification Methods, Datasets, Text Representation, Feature Extraction

1. INTRODUCTION

In today's information-driven world, the exponential growth of textual data has posed significant challenges in managing, organizing, and extracting meaningful insights from this sea of information. The need to efficiently categorize text documents has never been more crucial. Text classification plays a pivotal role in addressing this challenge, facilitating applications ranging from information retrieval and sentiment analysis to spam filtering and content recommendation. In the era of big data, it's not just the quantity of data that's overwhelming; it's also the diversity of sources and formats. With the advent of social media, blogs, research papers, and other digital platforms, text data now spans a wide spectrum of genres, languages, and modalities. This evolution necessitates the continuous development of novel methods and approaches to effectively classify this expanding corpus of textual information [12].

Text classification is one of the most fundamental challenges in the field of Natural Language Processing (NLP). Its significance is underlined by the fact that it often serves [70] [98] as a prerequisite for more complex NLP tasks such as machine translation, text summarization, and question-answering systems [7].

Text classification modules are components of numerous real-world applications, and they entail not only the core classifier responsible for mapping input data to specific classes but also a pipeline composed of sequential steps. There is no consensus on a universally accepted set of constituent stages of this pipeline, as researchers emphasize different aspects. For instance, within the work of [113], the text classification pipeline encompasses stages such as data collection, data analysis for labeling, feature construction and weighting, feature selection and projection, training of a classification model, and solution evaluation. In a related vein, [38] delineates a slightly distinct yet conceptually akin series of phases, including preprocessing, text representation, training of a classification model, and evaluation. Conversely, [62] propounds a distinct framework, encompassing document representation, feature selection, training of a classification model, and evaluation of a text classifier.

Within the framework of this comprehensive literature review, we aim to harmonize these disparate views by aggregating all phases relevant to text preparation and formalizing this central stage as «data preprocessing». At the same time, we recognize the intrinsic importance of the stages of training a classification model and conducting evaluations.

It is noteworthy that a significant proportion of the review studies in text classification have a relatively narrow focus, often focusing exclusively on the intrinsic properties of text classification models or delving into individual components within the classification process. For example, the studies by [147] and [159] are particularly specialized, closely examining the intricacies of classification algorithms and providing valuable insights into their merits and limitations. Conversely, the work of [104] omits any specific mention of the algorithms used for classification, drawing attention to other facets of the process. Of particular interest, [126] and [1] devote their attention to the area of feature selection methods, highlighting techniques for optimizing classification results. In stark contrast, the study by [76] abstains from any discourse on the underlying data and datasets used in their research.

The main objective of this study is to surpass the typical limitations seen in previous reviews on text classification. Our goal is to provide a thorough and extensive examination of the entire text classification pipeline, instead of limiting our investigation solely to classification algorithms or individual pipeline stages. This review will evaluate the technologies used for the process of text classification, including data preparation, processing, training classification models, and subsequent evaluations. Furthermore, this review will accentuate the role of datasets in the domain of text classification. Our objective is to conduct a systematic evaluation of the technologies used for the text classification process.

This systematic literature review's main contributions can be outlined as follows:

- Analyzing 110 scientific articles published between 2013 and 2022, this review extends existing research.
- By following established literature review protocols, it guarantees a trustworthy and replicable examination.
- The review catalogues and evaluates technologies used in the analyzed articles, including classification algorithms, feature engineering and text representation techniques.
- The review provides detailed listings of datasets used in the articles.

This paper is organized as follows: Section 2 summarizes the architecture of systems for text classification and provides information about the main elements of the text classification process, presents a description of data preparation, existing classification methods and classification evaluation. The methodology of the review, including the research questions, is detailed in Section 3. Section 4 presents the analysis of the main findings according to the research questions. The outcome of the entire review, taking into account the opinions of the authors, can be seen in Section 5.

2. BACKGROUND

This background section introduces the main concepts directly related to the topic of this review: Section 2.1 discusses the basic concept and architecture of the text document classification process. Section 2.2 presents information on existing technologies in the field of text data preparation. Section 2.3 presents the main types of architectures and algorithms for text classification.

2.1. Architecture of Text Classification Systems

Text classification, is a fundamental task in the field of NLP and text mining. Its primary objective is to automatically categorize or assign labels to a given text document into predefined categories or classes based on its content [75]. A machine learning model or algorithm typically performs this classification. Here are the key components and concepts related to text document classification [42]:

- Text classification involves assigning a document to one or more predetermined categories.
- The process typically requires a dataset (D) of labeled training documents (d_1, d_2, ..., d_n), each with corresponding class labels (C_1, C_2, ..., C_n). These labels serve as the ground truth to train a classification model.
- The aim of text classification is to create a model for assigning the correct class label to a new, unclassified document (d) automatically. This model learns from training data patterns and features that enable predictions about the class of new documents.
- There are two primary types of text classification: single and multi-label. In single label classification, each document is assigned to only one class. It is assumed to belong exclusively to a single category. A document in multi-label classification can belong to multiple classes at the same time. For example, a news article may be classified as both «politics» and «sports» if it covers topics from both categories.

A text classification pipeline is a structured sequence of processes designed to transform unstructured text data into useful information. As stated in the introduction we use tripartite division encompassing preparation, classifier training, and result evaluation.

The initial and fundamental step of the text classification pipeline involves preparing the text [150], which is an essential process for transforming unorganized textual data into an easily comprehendible and analyzable format. The importance of this phase cannot be overstated since the effectiveness and importance of the consequent analysis and model training are directly dependent on the meticulousness of text preparation.

The second phase of the text classification pipeline is dedicated to classifier training. A diverse array of machine learning algorithms, is contemplated in accordance with the nature of the text classification task. Armed with feature-extracted and labeled data, the selected classifier undergoes training to discern underlying patterns and relationships within the data. The ultimate objective is to endow the model with the ability to make accurate predictions on novel, unlabeled textual inputs.

The culmination of the text classification pipeline lies in result evaluation, a quintessential phase to gauge the efficacy and generalizability of the trained classifier. Evaluation metrics such as precision, recall, F1-score, and accuracy are employed to assess the model's performance [38]. These metrics provide a comprehensive view of the classifier's ability to correctly categorize text inputs.

Figure 2.1 shows a graphical representation of the text classification process.

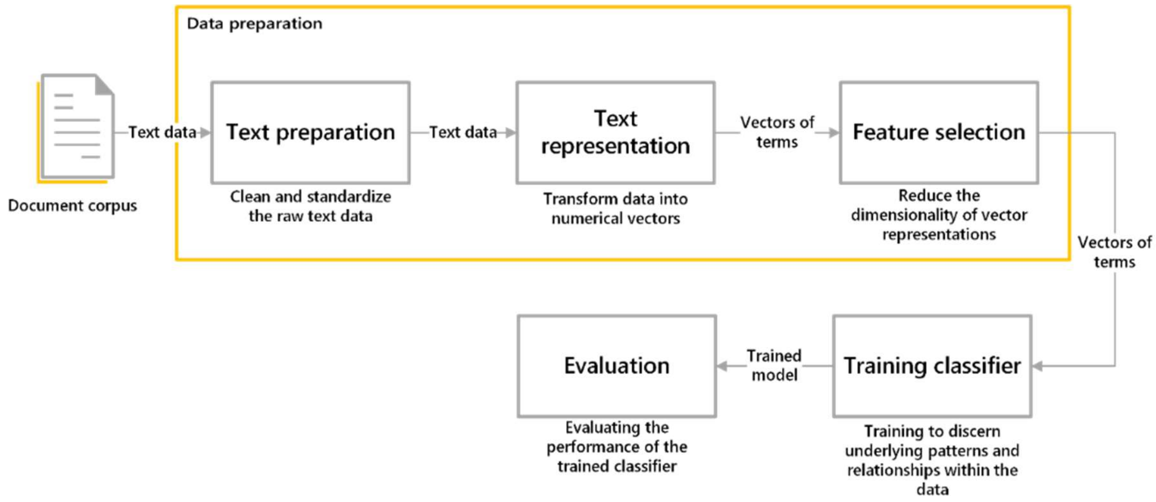

Fig. 2.1. Stages of text classification process

2.2. Data Preparation

The text preparation phase in a text classification pipeline is a crucial step that involves transforming raw, unstructured textual data into a structured and analyzable format. This phase encompasses several key components: text preprocessing, text representation and feature selection.

2.2.1 Text Preprocessing

The text preparation phase in a text classification pipeline is a crucial step that in-volves transforming raw, unstructured textual data into a structured and analyzable format. This phase encompasses several key components: text preprocessing, text representation and feature selection.

Text preprocessing is the initial step in text preparation. Its objective is to clean and standardize the raw text data to ensure consistency and remove unnecessary noise. Research papers ([150] [43] [115]) agree on important aspects of text preprocessing, such as stopword removal, tokenization, stemming and lemmatization.

Stopwords are common words like «the», «and», «in», which appear frequently in text but typically do not carry significant information for classification tasks. Removing stopwords reduces dimensionality and can improve model performance.

Tokenization involves breaking down the text into individual units, typically words or subwords, known as tokens. The process of tokenization plays a fundamental role in text analysis and classification. It allows the model to treat each token as a separate entity. Tokenization can be achieved through:

After tokenization, stemming and lemmatization are employed to reduce words to their base or root forms. The purpose of these processes is to standardize variations of words and reduce dimensionality.

Stemming is a more aggressive approach that involves removing prefixes and suffixes from words to reduce them to their root form [57]. For example, «running» and «runner» would both be stemmed to «run».

Lemmatization, on the other hand, is a more sophisticated technique that considers the context and grammar of words. It converts words to their base or dictionary forms, known as lemmas. For instance, «ran» would be lemmatized to «run» and «better» would be lemmatized to «good». Lemmatization often results in more meaningful and linguistically accurate base forms.

2.2.2 Text Representation

To enable mathematical operations on text data, it is often transformed into numerical vectors. This process, known as word representation, is essential for machine learning models. Two common methods for word representation are N-gram based models and Word Embedding Models.

N-gram based models represent a classical approach to text analysis, particularly in the context of representing words and sequences of words [38]. These models extract linguistic information by analyzing the frequency and co-occurrence of contiguous sequences of N words, where N represents the desired context window size. The TF-IDF score for a term t in a document d within a corpus is calculated using two key components: Term Frequency (TF) and Inverse Document Frequency (IDF) [65]. The TF component measures how frequently a term appears in a document, while the IDF component quantifies the importance of the term across the entire corpus. The overall TF-IDF score is a product of these two components:

$$\mathrm{TF\text{-}IDF}(\mathrm{t}, \mathrm{d}, \mathrm{D}) = \mathrm{TF}(\mathrm{t}, \mathrm{d}) \times \mathrm{IDF}(\mathrm{t}, \mathrm{D}) \quad (2.1)$$

where, $\mathrm{TF}(\mathrm{t}, \mathrm{d})$ represents the Term Frequency of term t in document d, which is often computed as the raw count of term occurrences or using normalized variants such as the sublinear or logarithmic TF. $\mathrm{IDF}(\mathrm{t}, \mathrm{D})$ represents the Inverse Document Frequency of term t in the corpus D, calculated as:

$$\mathrm{IDF}(\mathrm{t}, \mathrm{D}) = \log\left(\frac{\mathrm{N}}{\mathrm{n_t}}\right) \quad (2.2)$$

where N is the total number of documents in the corpus, $\mathrm{n_t}$ is the number of documents in which term t appears at least once.

Word Embedding Models represent a paradigm shift in word representation by encoding words as dense vectors in continuous vector spaces [38]. This approach has gained prominence due to its ability to capture semantic relationships between words, enabling advanced natural language understanding. This dense representation enables the capture of semantic information. Word embeddings can capture semantic relationships between words. For example, vector arithmetic operations like «king – man + woman» can yield the vector representation of «queen», illustrating the ability to infer word relationships [15]. Word Embedding Models, such as Word2Vec [112], GloVe [125], and FastText [17], map words to continuous vector spaces where words with similar meanings have similar vector representations.

Word2Vec is a word embedding technique that maps words from a large text corpus into continuous vector representations in a high-dimensional vector space [112]. The central idea behind Word2Vec is to learn distributed representations of words such that semantically similar words are mapped to nearby points in the vector space. Word2Vec operates under two primary architectures: Continuous Bag of Words (CBOW) and Skip-gram.

GloVe is a text representation model that aims to capture semantic relationships between words in a corpus by learning distributed word vectors [125]. It operates on the principle that word vectors should encode word co-occurrence information effectively. The core idea behind GloVe is to factorize a word co-occurrence matrix to learn word embeddings.

FastText is a text representation model developed by Facebook AI Research [17], designed to handle text classification efficiently. It extends the idea of Word2Vec by incorporating subword information. Instead of representing words as fixed vectors, FastText represents words as bags of character n-grams.

2.2.2 Feature Selection

Feature selection is the final step in the text preparation phase. In document classification, textual data is transformed into high-dimensional vectors through basic representation models, as previously described in section 2.2.2. While these high-dimensional vector representations capture the essence of the documents, they also introduce several inherent challenges – curse of dimensionality, sparse data and overfitting. To mitigate these challenges, feature selection techniques are employed to reduce the dimensionality of vector representations while preserving essential information [153]. Feature selection aims to identify and retain the most relevant features (terms or words) while discarding less informative ones.

Source [77] defines three categories of Feature Selection methods: filter, wrapper, and embedded. Filter methods like Chi-Squared Test (CHI), Information Gain (IG) and Mutual Information (MI) [83] are independent of specific machine learning algorithms. They evaluate features based on statistical metrics, such as chi-squared tests, information gain, or correlation coefficients, without considering the classification model. Selected features are typically based on their intrinsic relevance to the target class or document category.

The CHI algorithm is a statistical technique designed to assess the independence between a given feature (term) and class labels, effectively quantifying the significance of individual terms in a classification context. The CHI algorithm estimates a Chi-squared statistic for each feature, which is formulated as:

$$\chi^2 = \sum \frac{\left(\mathrm{O_{ij}} - \mathrm{E_{ij}}\right)^2}{\mathrm{E_{ij}}} \tag{2.3}$$

where, $\mathrm{O_{ij}}$ represents the observed frequency of feature i in class j, and $\mathrm{E_{ij}}$ denotes the expected frequency under the assumption of independence. A high CHI value indicates strong dependence between the feature and class labels, rendering it a valuable metric for feature selection.

IG method, deeply rooted in information theory, plays a pivotal role in the selection of relevant features for text classification. IG quantifies the reduction in uncertainty (entropy) regarding class labels achieved by incorporating a particular feature into the classification model. Essentially, it measures how much information a feature contributes to class discrimination. A higher IG signifies that the feature imparts substantial insights, enhancing the model's capability to distinguish between various classes.

MI measures the degree of dependency between a feature and class labels by assessing the shared information. Features with high MI values are considered informative as they exhibit a strong association with specific classes. MI, as a feature selection technique, aids in identifying features that contribute significantly to classification accuracy by quantifying the information shared between features and class labels.

Embedded methods incorporate feature selection into the model training process itself [83]. These methods optimize feature selection as an integral part of the learning algorithm. Examples include regularization techniques like L1 regularization (Lasso) [83] used in logistic regression and decision trees with built-in feature selection. Embedded methods aim to strike a balance between filter and wrapper methods, considering feature relevance and model performance simultaneously.

2.3 Classifier Training

In this section, we adopt a division of classification algorithms proposed by [7] to categorize them into two distinct groups: basic machine learning and deep learning.

2.3.1 Basic Machine Learning

Basic machine learning approaches are grounded in statistical principles and have been foundational in text classification tasks. These algorithms employ features derived from text data to make predictions. Examples of algorithms in this category include Naive Bayes (NB), SVM, Random Forest (RF), Logistic Regression (LR) and k-Nearest Neighbors (k-NN).

SVM [18] is a powerful and versatile machine learning algorithm commonly employed in text classification tasks. SVM operates by finding an optimal hyperplane that best separates data points belonging to different classes while maximizing the margin between them. In the context of text classification, documents are typically represented as high-dimensional feature vectors, where each dimension corresponds to a term or word.

NB [161] is a probabilistic classification algorithm widely used in text classification tasks. It operates on the principles of Bayes' theorem and conditional probability. One of the fundamental assumptions of NB is the «naive» assumption, which assumes that features (words or terms in the case of text classification) are conditionally independent, given the class label. In text classification, NB assigns class labels to documents based on the likelihood of observing certain terms given each class. The class with the highest probability is assigned as the predicted class label.

KNN [60] is a non-parametric classification algorithm used in text classification tasks. It operates based on the principle of proximity, where documents are assigned a class label based on the majority class among their k-nearest neighbors in the feature space. In text classification, KNN treats documents as points in a high-dimensional space, with each dimension representing a term or word. When classifying a new document, KNN identifies the k most similar documents from the training dataset, typically based on distance metrics like Euclidean distance or cosine similarity. The class label is then determined by a majority vote among these neighbors.

LR [39] - linear classification algorithm for text classification. It models the relationship between the binary or categorical dependent variable (class labels in text classification) and one or more independent variables (features or terms in text documents) using a logistic function. In text classification, LR assigns class probabilities to documents based on their features. It models the probability that a document belongs to a particular class given its feature values.

RF [56] is an ensemble learning algorithm used in text classification tasks. It combines multiple decision trees to improve classification accuracy and robustness. RF constructs an ensemble of decision trees, where each tree is trained on a bootstrapped subset of the training data and a random subset of features. The final class prediction is determined by aggregating the predictions of individual trees, often through majority voting.

2.3.2 Deep Learning

Deep learning represents a transformative paradigm in text classification, driven by the advancement of neural network architectures. Deep learning models, with their ability to automatically learn hierarchical representations, have excelled in capturing complex textual patterns.

CNNs, initially designed for image processing, have been adapted effectively for NLP tasks [58]. In text classification, CNNs utilize convolutional filters of varying sizes to scan and extract local features, such as n-grams, from the input text. These features are then processed hierarchically through multiple layers of the network to capture higher-level abstractions. This approach can effectively identify patterns and representations in text data, making it valuable for various text classification tasks.

RNNs, specifically Long Short-Term Memory (LSTM) [48] and Gated Recurrent Unit (GRU) [25] variants, are adept at modeling sequential data. RNNs are recurrent in nature, allowing them to process sequences of data by maintaining internal states and utilizing past information to influence future predictions [52]. They are particularly effective in tasks that involve sequential dependencies, such as text generation and sentiment analysis.

LSTM comprises recurrent cells with memory gates that enable the retention of crucial information over extended sequences. It tackles the vanishing gradient problem that plagues traditional RNNs, making it well-suited for tasks where context from distant words is essential. The LSTM cell consists of three primary gates: the input gate, the forget gate, and the output gate. These gates regulate the flow of information through the cell, allowing it to capture long-range dependencies in the text.

GRUs, like RNNs, are recurrent networks, but they incorporate gating mechanisms that enable better handling of long-range dependencies and mitigate the vanishing gradient problem. This makes them suitable for text classification tasks that involve capturing contextual information.

Transformer-based models [151], exemplified by the BERT (Bidirectional Encoder Representations from Transformers) architecture, have revolutionized text classification. BERT and its variants leverage attention mechanisms to capture contextual information, achieving state-of-the-art results in various NLP tasks, including document classification and question answering.

3. SYSTEMATIC LITERATURE REVIEW

The primary objective of this literature review is to systematically collect, analyze, and synthesize scholarly articles within the realm of text classification, specifically focusing on the diverse technologies employed in this domain. This comprehensive exploration aims to provide readers with a comprehensive understanding of the evolving landscape of text classification systems and the pivotal role of technological components therein. This review adheres to the well-established procedures outlined in Kitchenham's Systematic Literature Review protocol [74].

3.1 Research Questions

In pursuit of a comprehensive analysis, we have formulated two key research questions to guide this literature review:

R1. What components have been used in classification systems?

This primary research question directs our attention toward the diverse components and technologies embedded within text classification systems. We aim to categorize and explain the different components of these systems, including preprocessing techniques, feature selection methods, and the classification algorithms.

R2. What data (dataset) was used to classify the text in the works under study?

Our secondary research question examines the basic datasets employed in the analyzed works. We endeavor to identify and describe the datasets used for these classification systems' training and evaluation.

3.2 Constructing a Literature Review

The first critical step in constructing this literature review was the selection of electronic databases housing scholarly articles. To ensure comprehensive coverage of relevant research, five prominent databases were chosen as primary sources: IEEE Xplore, ACM Digital Library, Springer, ScienceDirect, and Google Scholar. These databases were selected due to their extensive repositories of academic publications across various domains, including artificial intelligence and NLP.

Given the focus of this review on text classification and the associated technologies, a comprehensive list of relevant keywords was compiled. These keywords include terms such as "Text classification techniques", "Document classification techniques", "Text categorization", "Document categorization", "Text classification", and "Text classification applications".

Leveraging the compiled keywords, a structured query was formulated for each of the selected electronic databases. An example query used is: "(((Text OR Texts OR Document OR Documents) AND (Categorization OR Classification OR Categorize OR Classify OR Categorizing OR Classifying OR Classifier)) AND (Techniques OR Applications))".

The period for the selection of articles spanning from 2013 to 2022 was strategically chosen to encapsulate a decade of scholarly research. This timeframe not only reflects a substantial period of academic investigation but also coincides with the exponential growth of research related to artificial intelligence, a prevailing trend in contemporary academia.

3.3 Literature Selection Process

The literature selection process went through several stages to guarantee the inclusion of both relevant and informative sources. At the outset, a search of designated electronic databases generated a substantial number of results – 9435. Following this, a methodical selection process was performed to enhance the quality of this population. The process involved assessing titles and keywords for relevance to the review's objectives while also removing duplicates. Ultimately, 243 sources were identified as potentially relevant.

To further narrow down the selection, we manually analyzed the abstracts and introductions of 243 sources based on a defined criterion. The presence of a classifier and a real-world experiment in the articles was the key inclusion criterion to ensure that the selected sources made a significant empirical contribution to the objectives of the review, rather than just being review articles. 110 research articles were selected for in-depth analysis during this literature review. Table 3.1 displays the respective number of articles for each scholarly database.

Table 3.1. Destitution articles for each scholarly database

Database	Potentially relevant results	Relevant results
IEEE Xplore	68	37
Science Direct	62	25
ACM Digital Library	47	19
Springer	37	16
Google Scholar	29	13

4. LITERATURE REVIEW RESULTS

In the results section, we present a comprehensive analysis of the 110 sources obtained from a thorough examination of five electronic databases. Section 4.1 pro-vides an overarching view of the literature review's outcomes, including the distribution of sources by publication year and classification of each source based on re-search focus. Section 4.2 delves into the outcomes of the investigation pertaining to the first research question, which centers on the technological aspects of text classification. This subsection is further divided into three sub-sections, each dedicated to elucidating the findings related to specific stages of the text classification process: data preparation, classifier training, and result evaluation. These sub-sections offer detailed insights into the myriad techniques and methods employed at each stage of the text classification pipeline. Lastly, Section 4.3 concentrates on the outcomes of the study concerning the second research question, focusing on the foundational datasets used in text classification research.

4.1 Literature Search Results

A graphical representation (Fig. 4.1) depicting the distribution of sources by publication year within the selected 10-year research period (2013-2022) has been compiled. Notably, this diagram reveals a significant surge in articles related to text classification in the year 2019, marking the pinnacle of research interest within the chosen timeframe. However, it's noteworthy that subsequent years display a gradual decline in scholarly output, a trend that can be attributed, at least in part, to the global impact of the coronavirus pandemic.

As elucidated in Section 3, the deliberate selection of a 10-year span aimed to spotlight the burgeoning interest in methods and approaches to text classification rooted in neural networks, particularly deep learning models. To scrutinize this evolution, a year-wise analysis was conducted, distinguishing between sources focusing on basic machine learning algorithms and those centered on deep learning.

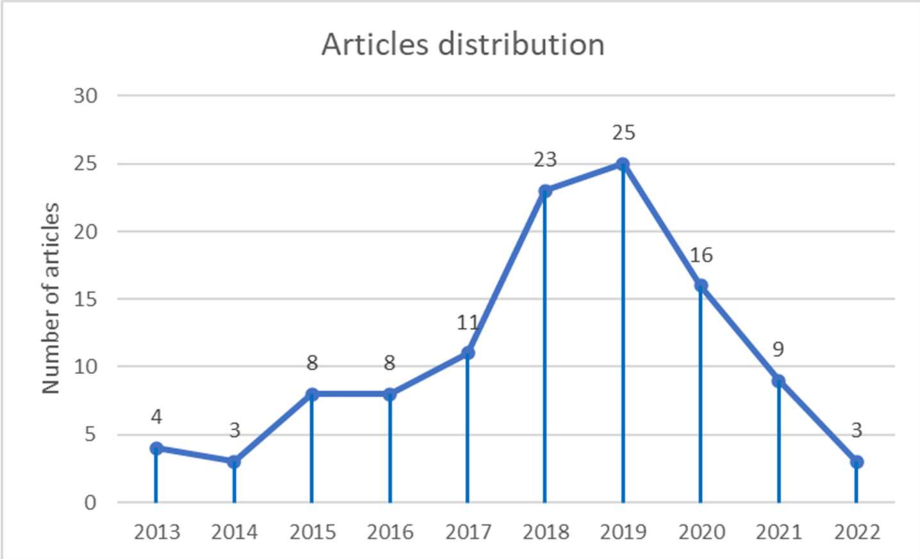

Fig. 4.1. Publishing summary from 2013 to 2022

Figure 4.2 provides a compelling narrative. It indicates that the emergence of interest in neural networks began around 2015, steadily gaining momentum. By 2019, deep learning and neural networks had reached their zenith, signifying a transformative shift in the field. Notably, the data suggests that this enthusiasm remains unabated, with deep learning approaches increasingly supplanting basic algorithms as the preferred choice for text classification tasks. This trajectory underscores the enduring relevance and potency of neural networks in text classification research.

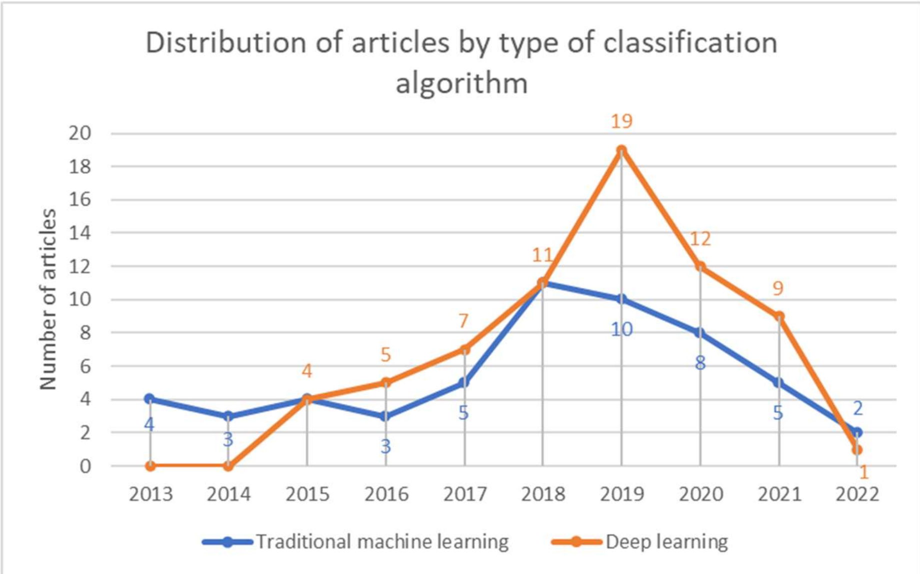

Fig. 4.2. Distribution of articles by type of classification algorithm from 2013 to 2022

Although the research theme consistently relates to text classification, it is important to note that various aspects of this field are often studied. However, all reviewed studies share an experimental component that involves creating and assessing a text classifier. The objectives of the research can be expressed by either focusing on the text classification problem or conducting a specialized study on the critical preparatory steps, including data preprocessing, text representation, and feature selection. The figure 4.3 reflects these nuances: most studies (83.64%) concentrate on the central issue of text classification or classifier development. A few works ([54] [55] [67] [89] [162] [96]) tackle text representation. In addition, several sources ([120] [171] [59] [14] [139] [90] [124] [149] [79] [129] [142] [130]) are dedicated to investigating feature selection techniques in-depth.

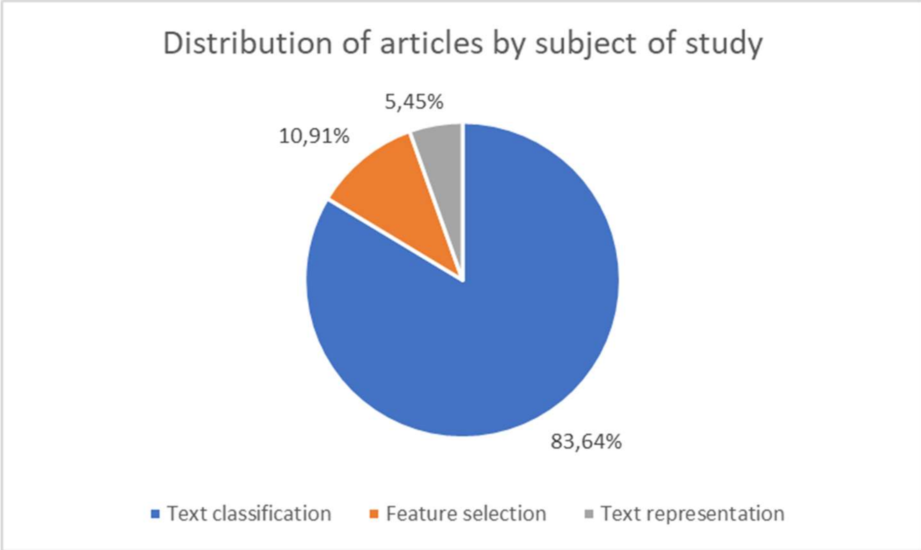

Fig. 4.3. Distribution of all articles by subject of study

4.2 RQ1 Results

4.2.1 Text Representation

While researching the sources, we compiled a list of text representation algorithms utilized in text classification tasks. To present our findings, table 4.1 displays the frequency of usage for particular text representation schemes. Among the total number of references, Word2Vec and TF-IDF are the two main methods, with respect to the total number of text representation algorithms in use, they are used in 48 percent of them.

Table 4.1. Text representation algorithms

Algorithm	Percentage	References
TF-IDF	24.5% (27 of 110)	[161] [56] [67] [89] [171] [14] [124] [149] [142] [130] [72] [87] [102] [157] [156] [107] [170] [165] [111] [24] [158] [22] [19] [8] [128] [13] [160]
Word2Vec	23.6% (27 of 110)	[54] [67] [96] [72] [168] [163] [87] [127] [137] [82] [103] [102] [157] [11] [140] [156] [27] [85] [53] [135] [174] [47] [36] [45] [154] [169]
BoW	10.9% (12 of 110)	[56] [162] [14] [8] [103] [11] [3] [44] [28] [92] [69] [21]
GloVe	6.3% (7 of 110)	[54] [102] [103] [176] [177] [81] [152]
FastText	2.7% (3 of 110)	[103] [53] [166]
Doc2Vec	2.7% (3 of 110)	[30] [157] [92]

In the following, we briefly outline the text representation algorithms highlighted by the review.

In 6.3% of the sources, the BoW text representation model is used. The BoW model treats each document as an unordered set of words and represents it as a sparse vector, where each dimension corresponds to a unique term in the entire corpus.

Doc2Vec, also known as paragraph vectors [80], extends the concept of word embeddings to entire documents. It learns fixed-length vector representations for documents by treating them as continuous pieces of text. Doc2Vec employs a neural network architecture that combines word vectors with a document-specific vector.

Source [96] employs an algorithm founded upon Word2Vec, known as Towe (task-oriented word embedding), tailored explicitly for text classification tasks. It introduces a function-aware component, strategically accentuating the functional attributes of words within the embedding space.

Source [89] presents an enhanced TF-IDF algorithm that emphasizes the concentration of feature words within specific categories, the dispersion degree within a class, and the strength of association between a feature word and a class.

Source [142] introduces the tax2Vec approach, which integrates semantic information from taxonomies to construct enhanced semantic features for improving the performance of short text classification. This novel algorithm leverages a labeled or unlabeled corpus of documents and a taxonomy of words.

Source [162] introduces a novel approach termed the Attribute of Network Extended to Bag-of-Words (AEBoW) model. AEBoW seeks to enhance traditional BoW representations by incorporating attribute weights from a network structure.

4.2.2 Feature Selection

In the corpus of research articles reviewed, it is noteworthy that only a minority, specifically 15.45%, explicitly incorporate feature selection techniques within their text classification methodologies ([171] [59] [14] [139] [124] [149] [79] [129] [142] [130] [158] [3] [152] [51] [5] [6] [2]). Among the feature selection methods encountered, the Chi-Square (CHI) algorithm emerges as the most prevalent and widely adopted ([171] [124] [149] [129] [152] [51] [6]). Information Gain stands as the second most prevalent feature selection algorithm within the scope of this review ([139] [129] [142] [51] [5]). Sources [129] and [142] utilize MI as a feature selection algorithm.

Sources [129] and [51] employ Document Frequency (DF) as a feature selection algorithm. DF assesses the importance of a feature by quantifying the number of documents in which it appears.

Source [59] utilizes Normalized Difference Measure (NDM) as a feature selection algorithm. NDM quantifies the dissimilarity between term distributions in different classes.

Source [14] employs Genetic Algorithms (GA) as a feature selection technique. GAs are optimization algorithms inspired by the process of natural selection. They iteratively evolve a population of feature subsets to maximize classification performance.

Moreover, it is noteworthy that source [139] introduces a novel adaptation of the Information Gain algorithm, termed Maximizing Global Information Gain (MGIG). This innovative approach substantially mitigates computational complexity, reducing it from O(VK2) to O(VK), where V signifies the number of terms and K represents the number of terms.

Source [149] employs a trio of feature selection algorithms: Poisson Distribution, Relative Discrimination Criterion, and Discriminative Features Selection (DFSS). Poisson Distribution models feature occurrence in documents as a Poisson process [119]. Features with a significantly different Poisson distribution across classes are selected. Relative Discrimination Criterion evaluates feature importance based on the relative entropy between class-conditional feature distributions [131]. Features that exhibit higher relative discrimination are chosen.

Source [79] employs the Firefly Algorithm as a feature selection technique. The Firefly Algorithm is a nature-inspired optimization algorithm that models the flashing behavior of fireflies to solve optimization problems [164]. In the context of feature selection, it operates by iteratively selecting a subset of features that maximize a specific objective function.

Source [129] employs an array of feature selection algorithms, including Odd Ratio and NGL Correlation Coefficient, in addition to CHI, IG, and MI:

Source [158] employs Singular Value Decomposition (SVD) as a feature selection algorithm. SVD is a dimensionality reduction technique that factors the term-document matrix into singular values, providing a low-rank approximation [31]. It effectively captures the most significant latent semantic structures within the data.

Source [2] employs Pearson Correlation Coefficients as a feature selection algorithm. Pearson Correlation measures the linear relationship between two variables. In feature selection, it quantifies the degree of association between individual features and class labels.

4.2.3 Text Classification

In line with the classification made in Section 2.3 of this review, we will present the findings regarding text classification research by distinguishing between basic machine learning algorithms and deep learning algorithms. To provide an overview of the findings regarding basic machine learning algorithms, a table 4.2 has been generated.

Table 4.2. Basic machine learning algorithms

Algorithm	Percentage	References
SVM	29% (32 of 110)	[55] [67] [89] [120] [171] [59] [14] [139] [149] [142] [72] [87] [157] [107] [111] [158] [13] [160] [103] [11] [3] [44] [28] [69] [51] [2] [88] [33] [68] [71] [16] [46]
NB	20% (22 of 110)	[161] [67] [120] [59] [14] [139] [124] [129] [157] [111] [158] [13] [3] [44] [28] [51] [2] [68] [71] [46] [122] [155] [141]
KNN	13,6% (15 of 110)	[162] [120] [14] [129] [157] [170] [111] [24] [13] [92] [2] [88] [71] [46] [116]
RF	8,1% (9 of 110)	[56] [120] [13] [44] [69] [68] [46] [122] [136]
LR	7,2% (8 of 110)	[120] [102] [157] [22] [13] [21] [68] [46]

Sources [89] and [87] uses a LinearSVM - a variant of SVM designed specifically for linearly separable data, where classes can be separated by a hyperplane [37]. LinearSVM employs a linear kernel function, which simplifies the classification boundary into a linear equation.

Source [88] introduces an innovative approach called SVM-KNN for text classification, which combines two distinct algorithms: SVM and KNN. The SVM-KNN algorithm aims to improve classification performance by leveraging the strengths of both SVM and KNN.

Source [33] introduces cbsSVM, a method that combines SVM with a learning strategy known as center-based similarity space learning [32].

Sources [44] and [122] employ a variant of Naive Bayes known as NB Network for text classification [35]. NB Network extends the basic NB algorithm by introducing a network structure, which allows for capturing dependencies among features (words) in the data.

Sources [161] and [46] used a Multi-Nominal Naive Bayes (Multi-NB). It is a specialized variant of Naive Bayes [106] designed for dealing with categorical data, which includes text data where the features represent the frequency or presence of terms.

Source [155] introduce a NBTree - a hybrid classification algorithm that combines the principles of decision trees with Naive Bayes classifiers. In this approach, a decision tree is constructed to make decisions based on the values of certain features.

Source [56] introduces an RF-based algorithm called Semantics Aware Random Forest (SARF) for text classification. SARF aims to enhance the performance of RF by considering semantics and the relevance of features in the classification process.

4.2.4 Deep Learning Algorithms

In analyzing the results of the literature review in Section 4.1, it was found that deep learning models have attracted considerable attention in the field of text classification. The results of the research on classifiers related to deep learning are shown in table 4.3.

Table 4.3. Deep learning algorithms

Algorithm	Percentage	References
LSTM	23.6% (26 of 110)	[54] [130] [102] [163] [137] [82] [103] [85] [53] [135] [174] [176] [81] [152] [141] [50] [175] [95] [114] [94] [138] [91] [23] [4]
CNN	20% (22 of 110)	[90] [19] [160] [82] [103] [11] [47] [45] [154] [169] [5] [68] [23] [20] [99] [100] [93] [173] [84] [134] [73] [10] [86]

RNN	10,9% (12 of 110)	[22] [8] [127] [103] [53] [45] [177] [4] [20] [99] [78] [132]
Transformer	8,1% (9 of 110)	[22] [103] [138] [23] [145] [41] [108] [61]

In the area of deep learning, it should be noted that the use of LSTM models has become one of the most prominent topics. LSTM, although technically an RNN model, has been widespread enough to merit a separate category because of its unique architecture and usefulness for analyzing sequential data. LSTM and its derived models represent the most commonly used classification algorithms among the sources reviewed. In addition, along with LSTM-based models, CNN models are the second most popular. RNN-based models and transformer-based models are not the least popular. Other learning models include graph neural networks (GNNs) ([165] [128] [29]) and hierarchical attention networks (HANs) ([142] [22] [127] [47] [36]). Sources ([177] [4] [132]) use GRU architecture for text classification.

Sources [130] [137] [152] [95] [114] [23] [4] employ the conventional LSTM architecture as a fundamental component of their text classification models.

Sources [54], [103] [102] [27] [176] [135] employ Bidirectional LSTM (BI-LSTM) models for text classification. BI-LSTM, an extension of standard LSTM, offers the ability to capture contextual information from both past and future states of a word within a sequence. BI-LSTM operates by employing two LSTM layers—one that processes the sequence from the beginning to the end (forward LSTM), and another that processes the sequence in reverse (backward LSTM).

Source [91] employs an innovative BiLSTM model called Attention-based Bi-directional LSTM with a Convolutional Layer (AC-BiLSTM). In AC-BiLSTM, the convolutional layer extracts higher-level phrase representations from word embedding vectors, and BiLSTM is used to access representations of preceding and following contexts.

Source [82] employs a Bi-LSTM-based model known as Bi-LSTM-CNN. In this architecture, the Bi-LSTM component captures contextual information from both directions, and this information is then fused through a CNN.

Source [85] introduces a model, known as Bidirectional Attention-based LSTM or ABLSTM, combines deep learning techniques with regular expression-based classification strategies to effectively categorize medical texts.

Source [174] employs a hybrid model called BRCAN (Bidirectional Recurrent Attention-based Convolutional Neural Network), which combines Bi-LSTM with a CNN, attention mechanisms, and word2vec embeddings.

Source [50] introduces the Universal Language Model Fine-tuning (ULMFiT) architecture, an extension of the AWD-LSTM model. This model, based on AWD-LSTM, represents a state-of-the-art approach to text classification. Furthermore, source [110] presents a modification to the LSTM model, focusing on a set of highly effective regularization strategies that can be seamlessly integrated into existing LSTM implementations. Source [141] and source [138] also leverage the ULMFiT model.

Additionally, source [175] introduces the C-LSTM architecture, which combines the strengths of CNNs and RNNs to enhance text classification. Source [81] similarly employs a model inspired by the C-LSTM architecture for text classification, further emphasizing the efficacy of this combined CNN and LSTM approach. Sources [47] and [100] uses a similar architecture, also combining CNN and RNN.

Source [163] introduces the rankLSTM classification model, which operates through a unified learning-ranking process. In rankLSTM, the arrangement of document labels is transformed based on a semantic tree, ensuring compatibility and consistency with the sequential learning LSTM framework. To address the challenge of variable label numbers within each document, rankLSTM employs Connectionist Temporal Classification (CTC) techniques [40], effectively mitigating the problem of error propagation.

Source [94] proposes an ASP-MTL (Adversarial Shared-Private Model for Multi-Task Learning) that incorporates LSTM layers. This architecture features a shared recurrent neural layer that engages in adversarial learning against a trained multi-layer perceptron, actively preventing it from making precise predictions regarding task types.

Moreover, sources [22] and [127] employ bidirectional GRU models with an attention mechanism. Bidirectional GRUs process input sequences in both forward and backward directions, enhancing their ability to capture contextual information. The attention mechanism allows the model to focus on important words or phrases within a sentence.

Source [53] introduces the Hierarchical Attention-based Recurrent Neural Network (HARNN), which leverages a hierarchical attention mechanism to model dependencies between different levels of a hierarchical structure. This top-down approach facilitates the capture of associations between texts and the hierarchical structure, making it effective for tasks that involve multi-level information.

Source [8] presents the DAL (Deep Active Learning) algorithm, which combines Deep Learning, based on RNN architecture, with Active Learning. This fusion of deep learning with active learning techniques enhances the model's performance while minimizing the need for extensive labeled data.

Source [45] employs a Bidirectional Recurrent Neural Network (BRNN). BRNNs have the advantage of processing variable-length sequences and maintaining access to both past and future information, which can be valuable for text classification tasks where contextual understanding is crucial.

Source [5] implements a Single-layer Multisize Filters Convolutional Neural Network (SMFCNN), which incorporates multiple filters of different sizes applied to various windows of text.

Source [86] presents a Convolutional Spatial Attention Neural Network (SVA-CNN) that utilizes a carefully designed training set of multiview representations, along with a combination of heterogeneous attention mechanisms and CNN-based operations. SVA-CNN automatically extracts and weights multiple granular and fine-grained representations, enhancing its performance in capturing complex textual features.

Most of these sources rely on the BERT (Bidirectional Encoder Representations from Transformers) ([22] [103] [138] [145] [108] [61]) model for text classification. BERT is a pre-trained transformer-based model that learns contextual word representations by considering the surrounding words in both directions (left and right) of a given word. This bidirectional context makes BERT particularly powerful in understanding context and semantics in text data.

Source [41] introduces the Multi-Scale Self-Attention and Multi-Scale Transformer, which combines multi-scale knowledge with self-attention mechanisms. This approach allows the model to extract rich and robust features from text data at different scales, enhancing its ability to understand context and semantics.

Source [61] presents LightXML, which adopts an end-to-end training approach with dynamic negative label sampling. LightXML employs generative cooperative networks to recall and rank labels, with the label recalling part generating negative and positive labels, and the label ranking part distinguishing positive labels.

Source [165] introduces Text GCN (Graph Convolutional Network), a method that leverages GNNs for text classification. It constructs a single large graph from an entire corpus, where words and documents serve as nodes. Text GCN utilizes graph convolutional operations to capture the structural dependencies and relationships between words and documents within this graph [128].

Source [29] introduces a text classification model based on GNNs known as HyperGAT (Hypergraph Attention Networks). HyperGAT is designed to capture high-order word interactions within a hypergraph representation of text data. It employs a dual attention mechanism to highlight crucial contextual information, enabling the learning of highly expressive text representations.

Source [36] introduces an architecture called HCAN (Hierarchical Convolutional Attention Networks), which combines the strengths of self-attention, capable of capturing linguistic relationships over long sequences like RNNs, with the efficiency of training found in CNNs.

4.2.5 Evaluation

During the analysis of the reviewed sources, the metrics employed by authors to evaluate the effectiveness of their text classification solutions were identified. Seven metrics were found to be used, with varying frequencies, and their usage are displayed in the figure 4.4. The most commonly employed metrics among researchers were Precision, Accuracy, Recall, and F-measure.

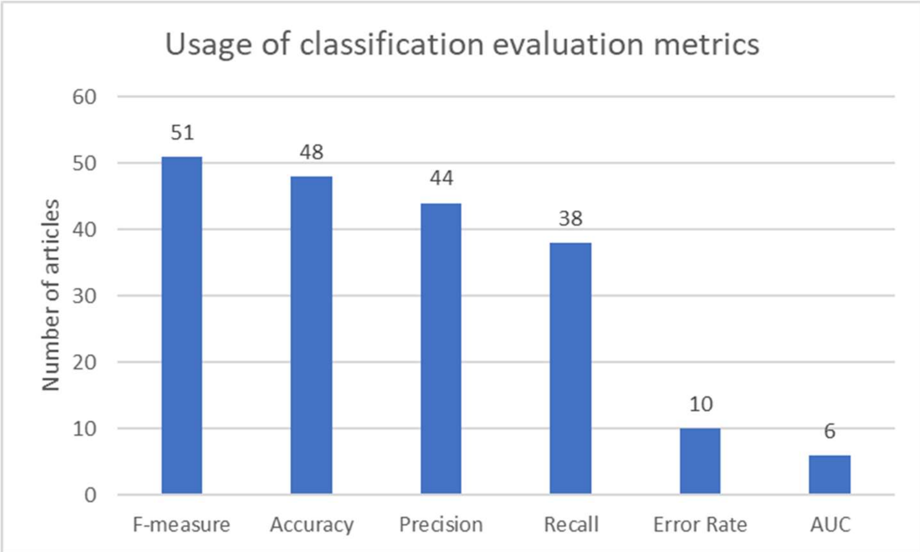

Fig. 4.4. Classification evaluation metrics

F-measure was utilized in 46% (51 of 110) percent of the analyzed sources. The F-measure, also known as the F1 Score, is a metric that combines both precision and recall into a single value, providing a balanced assessment of a classification model's performance. It is particularly useful when dealing with imbalanced datasets and is calculated as the harmonic mean of precision and recall. Furthermore, researchers often reported Micro, Macro, and Weighted Averages of the F1 Score to account for variations in class distribution.

Precision was employed in 43% (48 of 110) percent of the sources. Precision measures the accuracy of positive predictions made by a classifier, indicating the proportion of true positive predictions out of all positive predictions. It focuses on the precision of the model's positive predictions.

Accuracy was utilized in 40% (44 of 110) percent of the sources. Accuracy is a straightforward metric that measures the ratio of correctly predicted instances to the total instances in the dataset. It provides an overall assessment of a classifier's correctness.

Recall, also known as Sensitivity or True Positive Rate, was employed in 34% (38 of 110) percent of the sources. Recall measures the proportion of true positive predictions out of all actual positive instances in the dataset. It emphasizes the classifier's ability to correctly identify positive instances.

In contrast, fewer sources utilized metrics like Error Rate ([102] [82] [135] [177] [51] [141] [50] [114] [94] [145]) and Area Under the Curve (AUC) ([90] [72] [11] [53] [3] [2]). Error rate is a simple metric used to measure the accuracy of a classification model. It represents the proportion of incorrectly classified instances in a dataset. AUC is a metric used to evaluate the performance of binary classification models, particularly for tasks involving the calculation of Receiver Operating Characteristic (ROC) curves.

4.3 RQ2 Results

During the literature review, all datasets used by the researchers in their studies were organized and documented. These datasets have been compiled and are presented in accompanying Tables 4.4 and 4.5. Table 4.4 presents the distribution by domain.

Table 4.4. Dataset domain distribution

Domain	Number of datasets	Number of references
News	12	65
Reviews	10	59
Text	8	15
Medicine	10	15
Web Content	5	12
Questions	2	10
Academic papers	5	7
Legal	2	7
Articles	1	4
Documents	2	2
SMS	1	2

The table 4.5 shows the top 10 most popular datasets by number of mentions.

Table 4.5. 10 most popular dataset

Name	Domain	Source	Percentage	References
20 news group	News	[157]	15,5% (17 of 110)	[161] [162] [59] [14] [139] [130] [107] [158] [165] [81] [176] [33] [136] [78] [108] [29]
Reuters	News	[158]	13.6% (15 of 110)	[55] [162] [59] [14] [139] [124] [149] [129] [130] [107] [165] [81] [44] [136] [29]
IMDb	Reviews	[159]	11.8% (13 of 110)	[96] [135] [45] [141] [50] [145] [95] [114] [94] [91] [23] [108] [105]
MR	Reviews	[160]	10% (11 of 110)	[96] [165] [81] [154] [176] [94] [91] [173] [84] [86] [29]
SST	Reviews	[161]	9% (10 of 110)	[162] [96] [45] [154] [176] [175] [95] [91] [23] [78]
AG news	News	[162]	8.1% (9 of 110)	[102] [135] [154] [50] [4] [100] [86] [145] [66]
Amazon	Reviews	[164]	8.1% (9 of 110)	[54] [162] [36] [33] [93] [86] [108] [61] [66]
Dbpedia	Text	[165]	7.2% (8 of 110)	[135] [16] [50] [4] [100] [145] [108] [66]
Sogou News	News	[162]	6.3% (7 of 110)	[140] [27] [174] [100] [132] [145] [66]
Yelp	Reviews	[166]	6.3% (7 of 110)	[174] [36] [50] [100] [145] [66] [109]

However, for the sake of brevity and focused analysis, we will primarily delve into the datasets that do not fall within the realm of the top 10 most frequently employed datasets in the field of text classification. While the top 10 datasets provide valuable insights into the foundational and widely recognized benchmarks for text classification, our objective is to shed light on datasets that might not receive as much attention but are nonetheless significant for specific domains, research angles, or unique challenges. By examining these lesser-known datasets, we aim to broaden the scope of our analysis and capture a more comprehensive understanding of the diverse applications and contexts of text classification in the academic literature.

The EUR-LEX ([54] [22] [138] [93] [132] [61]) text collection is a collection of documents about European Union law [97]. It contains many several different types of documents, including treaties, legislation, case-law and legislative proposals, which are indexed according to several orthogonal categorization schemes to allow for multiple search facilities. The most important categorization is provided by the EUROVOC descriptors, which is a topic hierarchy with almost 4000 categories regarding different aspects of European law.

TREC ([154] [176] [177] [50] [91]) question dataset involves classifying a question into 6 question types [146]. TREC divides all questions into 6 categories, including 505 locations, human, entity, abbreviation, description and numeric. The training dataset contains 5452 labelled questions while the testing dataset contains 500 questions.

THUCNews ([8] [160] [82] [27]) corpus includes total of 10 news categories and total of 740,000 Chinese news texts, all in UTF-8 plain text format. It selects news from the ten categories of sports, finance, real estate, home, education, technology, fashion, politics, games and entertainment as experimental data.

The WebKB dataset ([162] [139] [44] [136]) contains 4,199 documents. Each document contains around 10,358 characters that are divided into around 2,918 words. The documents are webpages collected by the World-Wide Knowledge Base project of the CMU text-learning group. These pages were manually classified into seven different classes: student, faculty, staff, department, course, project, and other.

The Yahoo! Answers dataset ([174] [100] [145] [66]) consists of 10 categories including society and culture, science and mathematics, health, education and reference, computer and internet, sports, business and finance, entertainment and music, family and relationships, and politics and government. Documents include the title of the question, the background of the question and the best answer. There are 140,000 training samples and 5,000 test samples [172].

Three sources use datasets consisting of material from scholarly articles from the arXiv website. Source [54] uses a variation of the dataset consisting of 55,840 abstracts of computer science papers from 54 categories. Source [109] uses a variation of the dataset that contains 1 million abstracts of scientific papers, including the title and abstract of the article. Each paper has one primary category, e.g., cs.AI, stat.ML, etc., used as a class label. Source [135] uses a variation of the dataset consisting of 230,105 abstracts from 53 categories.

MEDLINE ([122] [136] [134]) and The Ohsumed ([165] [128] [29]) corpus is from the MEDLINE database [118], which is a bibliographic database of important medical literature maintained by the National Library of Medicine.

Elec ([135] [114]) consists of reviews of electronics products on Amazon and contains 200,000 reviews [64].

MIMIC-III ([103] [152]) dataset is an openly available dataset developed by the MIT Lab for Computational Physiology [63]. It comprises clinical notes, demographics, vital signs, laboratory tests and other data associated with 40,000 critical care patients.

2 sources use the New York Times news articles dataset. Source [109] independently constructs a dataset of 13,081 news articles using the New York Times API. This news corpus covers 5 supercategories and 25 subcategories. Source [47] uses the NYTimes, a large-scale corpus containing nearly every article published in the New York Times between January 01, 1987 and June 19th, 2007 [148], selecting articles with the NEWS tag.

Rotten Tomatoes ([114] [91]) consists of short snippets of movie reviews, for sentiment classification [121]. Rotten Tomatoes web site pages where reviews marked with «fresh» are labeled as positive, and reviews marked with «rotten» are labeled as negative

The SMS Spam Collection ([44] [46]) dataset is a public set of SMS labeled messages that have been collected for mobile phone spam research [143]. This dataset contains 5,574 English, real, and non-encoded messages, tagged as being legitimate (ham) or spam. Each SMS message contains around 82 characters that are divided into around 19 words.

Sources [88] and [20] use datasets based on information from Sohu, one of China's major websites. Source [20] uses a dataset that comes from the news data of 18 channels including domestic, international, sports, society and entertainment from June 2012 to July 2012, and provides URL and body information. Source [88] uses a dataset from Sohu laboratory.

Dataset uses in [13] is titled as «COVID Fake News Dataset» developed by [123] and published on Coronavirus Disease Research Community-Covid-19. Dataset contains 10202 COVID fake news shared all over social media platforms including Facebook, Instagram posts, and news on social media with the keywords COVID-19, coronavirus, and pandemic.

5. DISCUSSION

This section presents the discussion on all the RQs stated in Section 3.

5.1 What Components Have Been Used in Classification Systems?

Based on the comprehensive analysis conducted in this literature review, we can confidently assert that text classification has firmly established itself as a prominent and evolving problem within the domain of NLP. Researchers continue to devote considerable attention to this area, driven by its increasing significance and the diverse applications it encompasses.

One notable trend observed is the shift towards adopting cutting-edge technologies and novel approaches to address text classification challenges. As revealed in section 4.1, more and more researchers are applying neural network and deep learning technologies. This transformative trend extends beyond using deep learning solely as a classification tool; it significantly influences various facets of text classification. For instance, word representation algorithms like Word2Vec and GloVe, which are fundamental to text classification, rely on neural networks to learn word embeddings, thereby enhancing the quality of text representations.

Traditional text preprocessing techniques such as Bag of Words (BoW) and TF-IDF, which have historically underpinned text classification, still maintain their relevance, representing 44 percent of all analyzed sources. Researchers, however, are continually innovating within this space, often infusing these established techniques with neural network architectures ([89] [162] [142] [107]), thereby optimizing text preprocessing for modern classification tasks.

Feature selection for text classification remains an active research area, with researchers employing a variety of statistical algorithms in their classification pipelines ([171] [139] [79] [129]). Feature selection plays a pivotal role in enhancing the efficiency and effectiveness of classifiers.

While text representation and preprocessing are essential, classifiers themselves are the focal point of many studies. There is a great interest in recurrent neural networks, namely LSTM - according to the results of the study, it is on the basis of this variant of RNN that researchers most often proposed new methods, such as RankLSTM [163], AWD-LSTM [50], C-LSTM [175], CRNN [100], ASP-MTL [94], AC-BiLSTM [91], Bi-LSTM-CNN [82] and ABLSTM [85]. It is also worth noting that the use of multilayer networks gives researchers the opportunity to cross layers in different architectures, such as AC-BiLSTM, the convolutional layer extracts the higher-level phrase representations from the word embedding vectors and BiLSTM is used to access both the preceding and succeeding context representations. Bi-LSTM-CNN and NA-CNN-LSTM [99] present an architecture that integrates CNNs and RNNs.

The attention mechanism has also gained prominence in text classification ([54] [85] [53] [174] [36] [91] [73] [41] [29] [86]). Attention mechanisms enable models to focus on specific parts of input sequences, enhancing their ability to capture contextual information effectively.

Another pivotal development is the increasing utilization of transformer models in text classification ([103] [22] [138] [23] [145] [41] [108] [61]). The ascendancy of transformer models, epitomized by GPT and its variants, signifies the transformative potential of large-scale language models. Researchers recognize that these models are at the forefront of NLP, offering unmatched capabilities for understanding and generating human-like text. At present, the ChatGPT language model is the center of attention of the world community [34].

Despite the growing prevalence of neural network-based classification algorithms, basic algorithms like Naive Bayes and SVM continue to hold their ground - 50 percent of sources mention just basic algorithms. These basic methods offer several advantages, including simplicity, interpretability, and efficiency, which make them well-suited for certain classification tasks. Furthermore, for datasets with limited training examples, basic algorithms can often deliver competitive performance without the computational demands associated with deep learning [13].

In general, it can be observed that the evolving landscape of text classification is characterized by the fusion of deep learning, neural networks, basic methods and innovative approaches. Researchers are constantly pushing the boundaries of what is possible, driven by the need to develop efficient, accurate, and context-aware text classification solutions that meet the requirements of modern applications. The prevalence of deep learning, attention mechanisms, and transformational models emphasizes their key role in shaping the future of text classification.

5.2 What Data (Dataset) Was Used to Classify the Text in the Works Under Study?

Analyzing datasets used in text classification studies reveals interesting trends and preferences of researchers. Similar to source 11, we have given the 10 most popular datasets and also categorized them by domain. Among the most frequently used dataset types, news sets (20 news group, Reuters, AG news), covering areas such as sports, finance, health and politics, stand out, occurring in 65 of the 110 sources examined. These datasets are commonly used to solve multi-class categorization problems. In addition, datasets containing reviews, such as IMDb, MR, SST and Amazon, enjoy considerable popularity, mainly for binary categorization tasks where reviews are categorized as positive or negative.

However, this study also sheds light on less frequently employed datasets, often tailored to address specific domains or niche problems. One such example is the medicine field - 10 datasets were identified in the literature review and only 3 of them were mentioned 2 or more times (MEDLINE, Ohsumed, MIMIC-III), the remaining studies used unique datasets ([127] [21] [152] [173]). Similarly, datasets like EUR-LEX find their niche in legal domain applications ([54] [22] [138] [93] [61] [132]).

Another noteworthy trend is the use of question-and-answer datasets like TREC and Yahoo! Answers, featured in 10 out of 110 sources, particularly for multiclass classification, which often includes question categorization.

In summary, while popular datasets in news and reviews remain at the forefront of text classification research, the emergence of domain-specific datasets and specialized corpora highlights the evolving landscape of text classification, catering to a broader range of applications and research inquiries.

6. CONCLUSION

This paper presents a comprehensive literature review spanning the years 2013 to 2022, focusing on the technologies and datasets employed in the field of text classification. The review encompassed a total of 110 sources, all of which contained experimental data related to text classification processes. The study emphasized three fundamental stages in text classification: data preparation, classifier training, and result evaluation. Detailed statistics were compiled concerning the use of text representation and feature selection algorithms, as well as the various text classifiers adopted by researchers. Each technology integrated into the classification pipeline was meticulously examined within its respective source. Additionally, significant attention was dedicated to analyzing the diverse datasets utilized for text classification, with a particular emphasis on the less frequently employed ones. One prominent trend gleaned from this analysis is the increasing prevalence of deep learning-based technologies in the field. This study offers valuable insights into the evolving landscape of text classification, shedding light on the diverse array of technologies, approaches, and datasets employed by researchers.

REFERENCES

1. Abiodun, E. O., Alabdulatif, A., Abiodun, O. I., Alawida, M., Alabdulatif, A., et al. (2021). A systematic review of emerging feature selection optimization methods for optimal text classification: the present state and prospective opportunities, *Neural*

Computing and Applications, **33**(22), 15091–15118, https://doi.org/10.1007/S00521-021-06406-8

2. Adeleke, A. O., Samsudin, N. A., Mustapha, A. & Nawi, N. M. (2017). Comparative analysis of text classification algorithms for automated labelling of Quranic verses, *International Journal on Advanced Science, Engineering and Information Technology*, **7**(4), https://doi.org/10.18517/ijaseit.7.4.2198
3. Agnihotri, D., Verma, K. & Tripathi, P. (2018). An automatic classification of text documents based on correlative association of words, *Journal of Intelligent Information Systems*, **50**(3), https://doi.org/10.1007/s10844-017-0482-3
4. Aitken, K., Ramasesh, V. V., Garg, A., Cao, Y., Sussillo, D., et al. (2021). The geometry of integration in text classification RNNs, *Proc. of 9th International Conference on Learning Representations (ICLR 2021)* (Online).
5. Akhter, M. P., Jiangbin, Z., Naqvi, I. R., Abdelmajeed, M., Mehmood, A., et al. (2020). Document-Level Text Classification Using Single-Layer Multisize Filters Convolutional Neural Network, *IEEE Access*, **8**, 42689–42707, https://doi.org/10.1109/ACCESS.2020.2976744
6. Al-Tahrawi, M. M. & Al-Khatib, S. N. (2015). Arabic text classification using Polynomial Networks, *Journal of King Saud University – Computer and Information Sciences*, **27**(4), https://doi.org/10.1016/j.jksuci.2015.02.003
7. Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., et al. (2017). A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques. *arXiv*: 1707.02919v2, [Online]. Available: https://arxiv.org/abs/1707.02919v2
8. An, B., Wu, W. & Han, H. (2018). Deep active learning for text classification, *Proc. of 2nd International Conference on Vision, Image and Signal Processing (ICVISP 2018)* (New York, NY), 1–6, https://doi.org/10.1145/3271553.3271578
9. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., et al. (2007). DBpedia: A nucleus for a Web of open data, *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **4825 LNCS**, 722–735, https://doi.org/10.1007/978-3-540-76298-0_52/COVER
10. Baker, S. & Korhonen, A. (2017). Initializing neural networks for hierarchical multi-label text classification, *Proc. of SIGBioMed Workshop on Biomedical Natural Language Processing (BioNLP 2017)* (Vancouver, Canada), https://doi.org/10.18653/v1/w17-2339
11. Baker, S., Korhonen, A. & Pyysalo, S. (2016). Cancer Hallmark Text Classification Using Convolutional Neural Networks, *Proc. of the Fifth Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM2016)* (Osaka, Japan).
12. Banchs, R. E. (2021). Document Categorization. Text Mining with MATLAB®, 281–325. https://doi.org/10.1007/978-3-030-87695-1_11
13. Bangyal, W. H., Qasim, R., Rehman, N. U., Ahmad, Z., Dar, H., et al. (2021). Detection of Fake News Text Classification on COVID-19 Using Deep Learning Approaches, *Computational and Mathematical Methods in Medicine*, **2021**, https://doi.org/10.1155/2021/5514220
14. Belkarkor, S., Hafidi, I. & Nachaoui, M. (2023). Feature Selection for Text Classification Using Genetic Algorithm, *Lecture Notes in Networks and Systems*, **656 LNNS**. https://doi.org/10.1007/978-3-031-29313-9_7

15. Bengio, Y. (2009). Learning Deep Architectures for AI, *Foundations and Trends in Machine Learning*, **2**(1), 1–127, https://doi.org/10.1561/2200000006

16. Berardi, G., Esuli, A., Macdonald, C., Ounis, I. & Sebastiani, F. (2015). Semi-automated text classification for sensitivity identification. *Proc. of International Conference on Information and Knowledge Management (CIKM 2015)* (Boise , ID), https://doi.org/10.1145/2806416.2806597

17. Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. (2016). Enriching Word Vectors with Subword Information, *Transactions of the Association for Computational Linguistics*, **5**, 135–146, https://doi.org/10.1162/tacl_a_00051

18. Boser, B. E., Guyon, I. M. & Vapnik, V. N. (1992). Training algorithm for optimal margin classifiers, *Proc. of the Fifth Annual ACM Workshop on Computational Learning Theory* (Pittsburgh PA), 144–152, https://doi.org/10.1145/130385.130401

19. Boukil, S., Biniz, M., El Adnani, F., Cherrat, L. & Moutaouakkil, A. E. El. (2018). Arabic text classification using deep learning technics, *International Journal of Grid and Distributed Computing*, **11**(9), https://doi.org/10.14257/ijgdc.2018.11.9.09

20. Cai, J., Li, J., Li, W. & Wang, J. (2019). Deeplearning Model Used in Text Classification. *Proc. of 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)* (Chengdu, China), https://doi.org/10.1109/ICCWAMTIP.2018.8632592

21. Chai, K. E. K., Anthony, S., Coiera, E. & Magrabi, F. (2013). Using statistical text classification to identify health information technology incidents, *Journal of the American Medical Informatics Association*, **20**(5), https://doi.org/10.1136/amiajnl-2012-001409

22. Chalkidis, I., Fergadiotis, M., Malakasiotis, P. & Androutsopoulos, I. (2020). Large-scale multi-label text classification on EU legislation, *Proc. of 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)* (Florence, Italy), https://doi.org/10.18653/v1/p19-1636

23. Chen, H., Zheng, G. & Ji, Y. (2020). Generating hierarchical explanations on text classification via feature interaction detection, *Proc. of the Annual Meeting of the Association for Computational Linguistics* (Stroudsburg, PA), https://doi.org/10.18653/v1/2020.acl-main.494

24. Chen, Z., Zhou, L. J., Li, X. D., Zhang, J. N. & Huo, W. J. (2020). The Lao text classification method based on KNN, *Procedia Computer Science*, **166**, https://doi.org/10.1016/j.procs.2020.02.053

25. Cho, K., Van Merriënboer, B., Bahdanau, D. & Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches, *arXiv*: 1409.1259, [Online]. Available: https://arxiv.org/abs/1409.1259

26. Darst, B. F., Malecki, K. C. & Engelman, C. D. (2018). Using recursive feature elimination in random forest to account for correlated variables in high dimensional data, *BMC Genetics*, **19**(1), 1–6, https://doi.org/10.1186/S12863-018-0633-8/FIGURES/1

27. Deng, J., Cheng, L. & Wang, Z. (2021). Attention-based BiLSTM fused CNN with gating mechanism model for Chinese long text classification, *Computer Speech and Language*, **68**, https://doi.org/10.1016/j.csl.2020.101182

28. Desmet, B., & Hoste, V. (2018). Online suicide prevention through optimised text classification, *Information Sciences*, **439–440**, 61–78. https://doi.org/10.1016/j.ins.2018.02.014

29. Ding, K., Wang, J., Li, J., Li, D. & Liu, H. (2020). Be more with less: Hypergraph attention networks for inductive text classification, *Proc. of 2020 Conference on Empirical*

Methods in Natural Language Processing (EMNLP 2020) (Online), https://doi.org/10.18653/v1/2020.emnlp-main.399

30. Eken, S., Menhour, H. & Koksal, K. (2019). DoCA: A Content-Based Automatic Classification System over Digital Documents, *IEEE Access*, **7**, 97996–98004, https://doi.org/10.1109/ACCESS.2019.2930339

31. Fallucchi, F., & Zanzotto, F. M. (2009), Singular Value Decomposition for Feature Selection in Taxonomy Learning, *Proc. of International Conference Recent Advances in Natural Language Processing (RANLP 2009)* (Borovets, Bulgaria), 82–87.

32. Fei, G. & Liu, B. (2015). Social Media Text Classification under Negative Covariate Shift, *Proc. of the 2015 Conference on Empirical Methods in Natural Language Processing* (Lisbon, Portugal), 17–21.

33. Fei, G., & Liu, B. (2016). Breaking the closed world assumption in text classification, *Proc. of 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016)* (San Diego, CA), https://doi.org/10.18653/v1/n16-1061

34. Felten, E. W., Raj, M. & Seamans, R. (2023). How will Language Modelers like ChatGPT Affect Occupations and Industries?, *arXiv:* 2303.01157, [Online]. Available: https://arxiv.org/abs/2303.01157

35. Friedman, N., Geiger, D. & Goldszmidt, M. (1997). Bayesian Network Classifiers, *Machine Learning*, **29**(2–3), 131–163.

36. Gao, S., Ramanathan, A. & Tourassi, G. (2018). Hierarchical Convolutional Attention Networks for Text Classification, *Proc. of the Annual Meeting of the Association for Computational Linguistics* (Melbourne, Australia), https://doi.org/10.18653/v1/w18-3002

37. Gao, Y. & Sun, S. (2010). An empirical evaluation of linear and nonlinear kernels for text classification using support vector machines, *Proc. of 7th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010)* (Yantai, China), 1502–1505, https://doi.org/10.1109/FSKD.2010.5569327

38. Gasparetto, A., Marcuzzo, M., Zangari, A. & Albarelli, A. (2022). A Survey on Text Classification Algorithms: From Text to Predictions, *Information*, **13**(2), 83. https://doi.org/10.3390/INFO13020083

39. Genkin, A., Lewis, D. D., & Madigan, D. (2007). Large-scale bayesian logistic regression for text categorization, *Technometrics*, **49**(3), 291–304. https://doi.org/10.1198/004017007000000245

40. Graves, A. (2012). Connectionist Temporal Classification. In: A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks* (pp. 61–93), https://doi.org/10.1007/978-3-642-24797-2_7

41. Guo, Q., Qiu, X., Liu, P., Xue, X. & Zhang, Z. (2020). Multi-scale self-attention for text classification, *Proc. of 34th AAAI Conference on Artificial Intelligence (AAAI 2020)* (New York, NY), https://doi.org/10.1609/aaai.v34i05.6290

42. Gupta, V. & Lehal, G. S. (2009). A survey of text mining techniques and applications, *Journal of Emerging Technologies in Web Intelligence*, **1**(1), 60–76, https://doi.org/10.4304/JETWI.1.1.60-76

43. Gurusamy, V., & Kannan, S. (2014). Preprocessing Techniques for Text Mining, *International Journal of Computer Science & Engineering Technology*, **8**(06), 225–231.

44. HaCohen-Kerner, Y., Miller, D. & Yigal, Y. (2020). The influence of preprocessing on text classification using a bag-of-words representation, *PLoS ONE*, **15**(5), https://doi.org/10.1371/journal.pone.0232525
45. Hassan, A., & Mahmood, A. (2017). Efficient deep learning model for text classification based on recurrent and convolutional layers, *Proc. of 16th IEEE International Conference on Machine Learning and Applications (ICMLA 2017)* (Cancun, Mexico), https://doi.org/10.1109/ICMLA.2017.00009
46. Hassan, S. U., Ahamed, J. & Ahmad, K. (2022). Analytics of machine learning-based algorithms for text classification, *Sustainable Operations and Computers*, **3**, https://doi.org/10.1016/j.susoc.2022.03.001
47. He, Y., Li, J., Song, Y., He, M. & Peng, H. (2018). Time-evolving text classification with deep neural networks, *Proc. of International Joint Conference on Artificial Intelligence (IJCAI 2018)* (Stockholm, Sweden). https://doi.org/10.24963/ijcai.2018/310
48. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory, *Neural Computation*, **9**(8), 1735–1780, https://doi.org/10.1162/NECO.1997.9.8.1735
49. Rennie, J. (2008). Home Page for 20 Newsgroups Data Set, [Online]. Available: http://qwone.com/~jason/20Newsgroups/
50. Howard, J. & Ruder, S. (2018). Universal language model fine-tuning for text classification. *Proc. of 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (ACL 2018)* (Melbourne, Australia), https://doi.org/10.18653/v1/p18-1031
51. Hrala, M. & Král, P. (2013). Evaluation of the Document Classification Approaches, *Advances in Intelligent Systems and Computing*, **226**, 877–885, https://doi.org/10.1007/978-3-319-00969-8_86/COVER
52. Hu, H., Liao, M., Zhang, C. & Jing, Y. (2020). Text classification based recurrent neural network, *Proc. of 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC 2020)* (Chongqing, China), 652–655, https://doi.org/10.1109/ITOEC49072.2020.9141747
53. Huang, W., Chen, E., Liu, Q., Chen, Y., Huang, Z., et al. (2019). Hierarchical multi-label text classification: An attention-based recurrent network approach, Proc. of International Conference on Information and Knowledge Management (Boise, ID), https://doi.org/10.1145/3357384.3357885
54. Huang, X., Chen, B., Xiao, L., & Jing, L. (2019). Label-aware Document Representation via Hybrid Attention for Extreme Multi-Label Text Classification. arXiv: 1905.10070, [Online]. Available: https://arxiv.org/abs/1905.10070
55. Hung, V. T. (2020). Development of a document classification method by using geodesic distance to calculate similarity of documents, *Eastern-European Journal of Enterprise Technologies*, **4**, 2–106. https://doi.org/10.15587/1729-4061.2020.203866
56. Islam, M. Z., Liu, J., Li, J., Liu, L. & Kang, W. (2019). A semantics aware random forest for text classification, *Proc. of International Conference on Information and Knowledge Management* (Boise, ID), https://doi.org/10.1145/3357384.3357891
57. Ismailov, A., Abdul Jalil, M. M., Abdullah, Z., & Abd Rahim, N. H. (2016). A comparative study of stemming algorithms for use with the Uzbek language, *Proc. of 3rd International Conference on Computer and Information Sciences (ICCOINS 2016)* (Kuala Lumpur, Malaysia), https://doi.org/10.1109/ICCOINS.2016.7783180

58. Jacovi, A., Shalom, O. S., & Goldberg, Y. (2020). Understanding Convolutional Neural Networks for Text Classification, *arXiv*: 1809.08037, [Online]. Available: https://arxiv.org/abs/1809.08037

59. Javed, K. & Babri, H. A. (2017). Feature selection based on a normalized difference measure for text classification, *Information Processing and Management*, **53**(2), https://doi.org/10.1016/j.ipm.2016.12.004

60. Jiang, S., Pang, G., Wu, M. & Kuang, L. (2012). An improved K-nearest-neighbor algorithm for text categorization, *Expert Systems with Applications*, **39**(1), 1503–1509, https://doi.org/10.1016/J.ESWA.2011.08.040

61. Jiang, T., Wang, D., Sun, L., Yang, H., Zhao, Z. & Zhuang, F. (2021). LightXML: Transformer with Dynamic Negative Sampling for High-Performance Extreme Multi-label Text Classification, *Proc. of 35th AAAI Conference on Artificial Intelligence (AAAI 2021)* (Online), https://doi.org/10.1609/aaai.v35i9.16974

62. Jindal, R., Malhotra, R. & Jain, A. (2015). Techniques for text classification: Literature review and current trends, *Webology*, **12**(2), 1–28.

63. Johnson, A. E. W., Pollard, T. J., Shen, L., Lehman, L. W. H., Feng, M., et al. (2016). MIMIC-III, a freely accessible critical care database, *Scientific Data 2016*, **3**(1), 1–9, https://doi.org/10.1038/sdata.2016.35

64. Johnson, R. & Zhang, T. (n.d.). Semi-supervised Convolutional Neural Networks for Text Categorization via Region Embedding, *arXiv*: 1504.01255, [Online]. Available: https://arxiv.org/abs/1504.01255.

65. Jones, K. S. (2004). A statistical interpretation of term specificity and its application in retrieval, *Journal of Documentation*, **60**, 11–21.

66. Joulin, A., Grave, E., Bojanowski, P. & Mikolov, T. (2017). Bag of tricks for efficient text classification, *Proc. of 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)* (Valencia, Spain), https://doi.org/10.18653/v1/e17-2068

67. Kalra, V., Kashyap, I. & Kaur, H. (2022). Improving document classification using domain-specific vocabulary: hybridization of deep learning approach with TFIDF, *International Journal of Information Technology*, **14**(5). https://doi.org/10.1007/s41870-022-00889-x

68. Kamath, C. N., Bukhari, S. S. & Dengel, A. (2018). Comparative study between traditional machine learning and deep learning approaches for text classification, *Proc. of the ACM Symposium on Document Engineering* (Halifax, Canada), https://doi.org/10.1145/3209280.3209526

69. Karthikeyan, T., Sekaran, K., Ranjith, D., Vinoth kumar, V. & Balajee, J. M. (2019). Personalized content extraction and text classification using effective web scraping techniques, *International Journal of Web Portals*, **11**(2), https://doi.org/10.4018/IJWP.2019070103

70. Khalilian, M., & Hassanzadeh, S. (2019). Document classification methods, *arXiv*: 1909.07368v1, [Online]. Available: https://arxiv.org/abs/1909.07368v1

71. Khamar, K. (2013). Short Text Classification Using kNN Based on Distance Function, *International Journal of Advanced Research in Computer and Communication Engineering*, **2**(4).

72. Kim, M.-J., Kang, J.-S. & Chung, K. (2020). Word-Embedding-Based Traffic Document Classification Model for Detecting Emerging Risks Using Sentiment Similarity Weight, *IEEE Access*, **8**, 183983–183994, https://doi.org/10.1109/ACCESS.2020.3026585

73. Kim, Y. (2014). Convolutional neural networks for sentence classification, *Proc. of Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)* (Doha, Qatar), https://doi.org/10.3115/v1/d14-1181

74. Kitchenham, B. (2004). *Procedures for Performing Systematic Reviews*. Keele, UK: Keele University.

75. Korde, V. & Mahender, C. N. (2012). Text classification and classifiers: a survey, *International Journal of Artificial Intelligence & Applications*, **3**(2), https://doi.org/10.5121/ijaia.2012.3208

76. Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., et al. (2019). Text Classification Algorithms: A Survey, *Information 2019*, **10**(4), 150, https://doi.org/10.3390/INFO10040150

77. Kumar, V. (2014). Feature Selection: A literature Review, *The Smart Computing Review*, **4**(3), https://doi.org/10.6029/SMARTCR.2014.03.007

78. Lai, S., Xu, L., Liu, K. & Zhao, J. (2015). Recurrent convolutional neural networks for text classification, Proc. of the National Conference on Artificial Intelligence (Austin, TX), https://doi.org/10.1609/aaai.v29i1.9513

79. Larabi Marie-Sainte, S. & Alalyani, N. (2020). Firefly Algorithm based Feature Selection for Arabic Text Classification, *Journal of King Saud University – Computer and Information Sciences*, **32**(3), 320–328, https://doi.org/10.1016/J.JKSUCI.2018.06.004

80. Le, Q. & Mikolov, T. (2014). Distributed Representations of Sentences and Documents, *arXiv*: 1405.4053, [Online]. Available: https://arxiv.org/abs/1405.4053.

81. Leclair, A., Eberhart, Z. & McMillan, C. (2018). Adapting neural text classification for improved software categorization, *Proc. of the 2018 IEEE International Conference on Software Maintenance and Evolution (ICSME 2018)* (Madrid, Spain), https://doi.org/10.1109/ICSME.2018.00056

82. Li, C., Zhan, G. & Li, Z. (2018). News Text Classification Based on Improved Bi-LSTM-CNN, *Proc. of the 9th International Conference on Information Technology in Medicine and Education (ITME 2018)* (Hangzhou, China). https://doi.org/10.1109/ITME.2018.00199

83. Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature Selection: A Data Perspective. ACM Comput. Surv, 50. https://doi.org/10.1145/3136625

84. Li, S., Zhao, Z., Liu, T., Hu, R., & Du, X. (2017). Initializing convolutional filters with semantic features for text classification. EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings. https://doi.org/10.18653/v1/d17-1201

85. Li, X., Cui, M., Li, J., Bai, R., Lu, Z., & Aickelin, U. (2021). A hybrid medical text classification framework: Integrating attentive rule construction and neural network. Neurocomputing, 443. https://doi.org/10.1016/j.neucom.2021.02.069

86. Liang, Y., Li, H., Guo, B., Yu, Z., Zheng, X., Samtani, S., & Zeng, D. D. (2021). Fusion of heterogeneous attention mechanisms in multi-view convolutional neural network for text classification. Information Sciences, 548. https://doi.org/10.1016/j.ins.2020.10.021

87. Lilleberg, J., Zhu, Y., & Zhang, Y. (2015). Support vector machines and Word2vec for text classification with semantic features. Proceedings of 2015 IEEE 14th International

Conference on Cognitive Informatics and Cognitive Computing, ICCI*CC 2015. https://doi.org/10.1109/ICCI-CC.2015.7259377
88. Lin, Y., & Wang, J. (2014). Research on text classification based on SVM-KNN. Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS. https://doi.org/10.1109/ICSESS.2014.6933697
89. Liu, C. Z., Sheng, Y. X., Wei, Z. Q., & Yang, Y. Q. (2018). Research of Text Classification Based on Improved TF-IDF Algorithm. 2018 IEEE International Conference of Intelligent Robotic and Control Engineering, IRCE 2018. https://doi.org/10.1109/IRCE.2018.8492945
90. Liu, F., & Avci, B. (2020). Incorporating priors with feature attribution on text classification. ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference. https://doi.org/10.18653/v1/p19-1631
91. Liu, G., & Guo, J. (2019). Bidirectional LSTM with attention mechanism and convolutional layer for text classification. Neurocomputing, 337, 325–338. https://doi.org/10.1016/J.NEUCOM.2019.01.078
92. Liu, H., Burnap, P., Alorainy, W., & Williams, M. L. (2019). A Fuzzy Approach to Text Classification with Two-Stage Training for Ambiguous Instances. IEEE Transactions on Computational Social Systems, 6(2). https://doi.org/10.1109/TCSS.2019.2892037
93. Liu, J., Chang, W. C., Wu, Y., & Yang, Y. (2017). Deep learning for extreme multi-label text classification. SIGIR 2017 - Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. https://doi.org/10.1145/3077136.3080834
94. Liu, P., Qiu, X., & Huang, X. (2017). Adversarial multi-task learning for text classification. ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), 1. https://doi.org/10.18653/v1/P17-1001
95. Liu, P., Qiu, X., & Xuanjing, H. (2016). Recurrent neural network for text classification with multi-task learning. IJCAI International Joint Conference on Artificial Intelligence, 2016-January.
96. Liu, Q., Huang, H., Gao, Y., Wei, X., Tian, Y., & Liu, L. (2018). Task-oriented word embedding for text classification. COLING 2018 - 27th International Conference on Computational Linguistics, Proceedings.
97. Loza Mencía, E., & Fürnkranz, J. (2008). Efficient pairwise multilabel classification for large-scale problems in the legal domain. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5212 LNAI(PART 2), 50–65. https://doi.org/10.1007/978-3-540-87481-2_4/COVER
98. Lu, M., & Liang, P. (2017). Automatic classification of non-functional requirements from augmented app user reviews. ACM International Conference Proceeding Series, Part F128635, 344–353. https://doi.org/10.1145/3084226.3084241
99. Luan, Y., & Lin, S. (2019). Research on Text Classification Based on CNN and LSTM. Proceedings of 2019 IEEE International Conference on Artificial Intelligence and Computer Applications, ICAICA 2019. https://doi.org/10.1109/ICAICA.2019.8873454

100. Lyu, S., & Liu, J. (2021). Convolutional recurrent neural networks for text classification. Journal of Database Management, 32(4). https://doi.org/10.4018/JDM.2021100105
101. Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning Word Vectors for Sentiment Analysis (pp. 142–150). https://aclanthology.org/P11-1015
102. Marivate, V., & Sefara, T. (2020). Improving Short Text Classification Through Global Augmentation Methods. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12279 LNCS. https://doi.org/10.1007/978-3-030-57321-8_21
103. Mascio, A., Kraljevic, Z., Bean, D., Dobson, R., Stewart, R., Bendayan, R., & Roberts, A. (2020). Comparative analysis of text classification approaches in electronic health records. Proceedings of the Annual Meeting of the Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.bionlp-1.9
104. Maw, M., Balakrishnan, V., Rana, O., & Ravana, S. D. (2020). TRENDS AND PATTERNS OF TEXT CLASSIFICATION TECHNIQUES: A SYSTEMATIC MAPPING STUDY. Malaysian Journal of Computer Science, 33(2), 102–117. https://doi.org/10.22452/MJCS.VOL33NO2.2
105. McAuley, J., & Leskovec, J. (2013). Hidden factors and hidden topics: Understanding rating dimensions with review text. RecSys 2013 - Proceedings of the 7th ACM Conference on Recommender Systems, 165–172. https://doi.org/10.1145/2507157.2507163
106. Mccallum, A., & Nigam, K. (n.d.). A Comparison of Event Models for Naive Bayes Text Classification.
107. Meena, K., & Lawrance, R. (2019). An automatic text document classification using modified weight and semantic method. International Journal of Innovative Technology and Exploring Engineering, 8(12). https://doi.org/10.35940/ijitee.K2123.1081219
108. Meng, Y., Shen, J., Zhang, C., & Han, J. (2019). Weakly-supervised hierarchical text classification. 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019. https://doi.org/10.1609/aaai.v33i01.33016826
109. Meng, Y., Zhang, Y., Huang, J., Xiong, C., Ji, H., Zhang, C., & Han, J. (2020). Text classification using label names only: A language model self-training approach. EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference. https://doi.org/10.18653/v1/2020.emnlp-main.724
110. Merity, S., Shirish Keskar, N., & Socher, R. (2017). Regularizing and Optimizing LSTM Language Models.
111. Miao, F., Zhang, P., Jin, L., & Wu, H. (2018). Chinese News Text Classification Based on Machine Learning Algorithm. Proceedings - 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2018, 2. https://doi.org/10.1109/IHMSC.2018.10117
112. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (n.d.). Efficient Estimation of Word Representations in Vector Space. Retrieved September 14, 2023, from http://ronan.collobert.com/senna/

113. Mirończuk, M. M., & Protasiewicz, J. (2018). A recent overview of the state-of-the-art elements of text classification. Expert Systems with Applications, 106, 36–54. https://doi.org/10.1016/J.ESWA.2018.03.058

114. Miyato, T., Dai, A. M., & Goodfellow, I. (2017). Adversarial training methods for semi-supervised text classification. 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings.

115. Mohan, V. (2015). Preprocessing Techniques for Text Mining - An Overview.

116. Moldagulova, A., & Sulaiman, R. B. (2018). Document Classification Based on KNN Algorithm by Term Vector Space Reduction. 2018 18th International Conference on Control, Automation and Systems (ICCAS), 387–391.

117. Movie Review Data. (n.d.). Retrieved September 18, 2023, from http://www.cs.cornell.edu/people/pabo/movie-review-data/

118. MTI ML (Machine Learning Package). (n.d.). Retrieved September 18, 2023, from https://lhncbc.nlm.nih.gov/ii/tools/MTI_ML.html

119. Ogura, H., Amano, H., & Kondo, M. (2009). Feature selection with a measure of deviations from Poisson in text categorization. Expert Systems with Applications, 36(3), 6826–6832. https://doi.org/10.1016/J.ESWA.2008.08.006

120. Onan, A. (2016). An ensemble scheme based on language function analysis and feature engineering for text genre classification. Journal of Information Science, 44(1), 28–47. https://doi.org/10.1177/0165551516677911

121. Pang, B., & Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. ACL-05 - 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference.

122. Parlak, B., & Uysal, A. K. (2015). Classification of medical documents according to diseases. 2015 23rd Signal Processing and Communications Applications Conference, SIU 2015 - Proceedings, 1635–1638. https://doi.org/10.1109/SIU.2015.7130164

123. Patwa, P., Sharma, S., Pykl, S., Guptha, V., Kumari, G., Akhtar, M. S., Ekbal, A., Das, A., & Chakraborty, T. (2021). Fighting an Infodemic: COVID-19 Fake News Dataset. Communications in Computer and Information Science, 1402 CCIS. https://doi.org/10.1007/978-3-030-73696-5_3

124. Paudel, S., Prasad, P. W. C., & Alsadoon, A. (2018). Feature Selection Approach for Twitter Sentiment Analysis and Text Classification Based on Chi-Square and Naïve Bayes. In International Conference on Applications and Techniques in Cyber Security and Intelligence ATCI 2018 (Vol. 842, Issue 1).

125. Pennington, J., Socher, R., & Manning, C. D. (n.d.). GloVe: Global Vectors for Word Representation. 1532–1543. Retrieved September 14, 2023, from http://nlp.

126. Pintas, J. T., Fernandes, L. A. F., & Garcia, A. C. B. (2021). Feature selection methods for text classification: a systematic literature review. Artificial Intelligence Review 2021 54:8, 54(8), 6149–6200. https://doi.org/10.1007/S10462-021-09970-6

127. Qing, L., Linhong, W., & Xuehai, D. (2019). A Novel Neural Network-Based Method for Medical Text Classification. Future Internet 2019, Vol. 11, Page 255, 11(12), 255. https://doi.org/10.3390/FI11120255

128. Ragesh, R., Sellamanickam, S., Iyer, A., Bairi, R., & Lingam, V. (2021). HeteGCN: Heterogeneous Graph Convolutional Networks for Text Classification. WSDM 2021 -

Proceedings of the 14th ACM International Conference on Web Search and Data Mining. https://doi.org/10.1145/3437963.3441746
129. Rahamat Basha, S., & Rani, J. K. (2019). A Comparative Approach of Dimensionality Reduction Techniques in Text Classification. Engineering, Technology & Applied Science Research, 9(6). https://doi.org/10.48084/etasr.3146
130. Rehman, A., Javed, K., Babri, H. A., & Asim, N. (2018). Selection of the most relevant terms based on a max-min ratio metric for text classification. Expert Systems with Applications, 114. https://doi.org/10.1016/j.eswa.2018.07.028
131. Rehman, A., Javed, K., Babri, H. A., & Saeed, M. (2015). Relative discrimination criterion - A novel feature ranking method for text data. Expert Systems with Applications, 42(7), 3670–3681. https://doi.org/10.1016/J.ESWA.2014.12.013
132. Ren, F., & Deng, J. (2018). Background knowledge based multi-stream neural network for text classification. Applied Sciences (Switzerland), 8(12). https://doi.org/10.3390/app8122472
133. Reuters-21578 Text Categorization Test Collection. (n.d.). Retrieved September 18, 2023, from http://www.daviddlewis.com/resources/testcollections/reuters21578/
134. Rios, A., & Kavuluru, R. (2015). Convolutional neural networks for biomedical text classification: Application in indexing biomedical articles. BCB 2015 - 6th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics. https://doi.org/10.1145/2808719.2808746
135. Sachan, D. S., Zaheer, M., & Salakhutdinov, R. (2019). Revisiting LSTM networks for semi-supervised text classification via mixed objective function. 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019. https://doi.org/10.1609/aaai.v33i01.33016940
136. Salles, T., Gonçalves, M., Rodrigues, V., & Rocha, L. (2018). Improving random forests by neighborhood projection for effective text classification. Information Systems, 77. https://doi.org/10.1016/j.is.2018.05.006
137. Semberecki, P., & MacIejewski, H. (2017). Deep learning methods for subject text classification of articles. Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017. https://doi.org/10.15439/2017F414
138. Shaheen, Z., Wohlgenannt, G., & Filtz, E. (2020). Large Scale Legal Text Classification Using Transformer Models. https://arxiv.org/abs/2010.12871v1
139. Shang, C., Li, M., Feng, S., Jiang, Q., & Fan, J. (2013). Feature selection via maximizing global information gain for text classification. Knowledge-Based Systems, 54. https://doi.org/10.1016/j.knosys.2013.09.019
140. She, X., & Zhang, D. (2018). Text Classification Based on Hybrid CNN-LSTM Hybrid Model. Proceedings - 2018 11th International Symposium on Computational Intelligence and Design, ISCID 2018, 2. https://doi.org/10.1109/ISCID.2018.10144
141. Shleifer, S. (2019). Low Resource Text Classification with ULMFit and Backtranslation. https://arxiv.org/abs/1903.09244v2
142. Škrlj, B., Martinc, M., Kralj, J., Lavrač, N., & Pollak, S. (2021). tax2vec: Constructing Interpretable Features from Taxonomies for Short Text Classification. Computer Speech & Language, 65, 101104. https://doi.org/10.1016/J.CSL.2020.101104

143. SMS Spam Collection - UCI Machine Learning Repository. (n.d.). Retrieved September 18, 2023, from https://archive.ics.uci.edu/dataset/228/sms+spam+collection

144. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank (pp. 1631–1642). Association for Computational Linguistics. https://aclanthology.org/D13-1170

145. Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to Fine-Tune BERT for Text Classification? Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11856 LNAI. https://doi.org/10.1007/978-3-030-32381-3_16

146. Text REtrieval Conference (TREC) Data. (n.d.). Retrieved September 18, 2023, from https://trec.nist.gov/data.html

147. Thangaraj, M., & Sivakami, M. (2018). Text Classification Techniques: A Literature Review. Interdisciplinary Journal of Information, Knowledge, and Management, 13, 117–135. https://doi.org/10.28945/4066

148. The New York Times Annotated Corpus - Linguistic Data Consortium. (n.d.). Retrieved September 18, 2023, from https://catalog.ldc.upenn.edu/LDC2008T19

149. Uysal, A. K. (2018). On Two-Stage Feature Selection Methods for Text Classification. IEEE Access, 6. https://doi.org/10.1109/ACCESS.2018.2863547

150. Uysal, A. K., & Gunal, S. (2014). The impact of preprocessing on text classification. Information Processing & Management, 50(1), 104–112. https://doi.org/10.1016/J.IPM.2013.08.006

151. Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (n.d.). Attention Is All You Need.

152. Venkataraman, G. R., Pineda, A. L., Bear Don't Walk, O. J., Zehnder, A. M., Ayyar, S., Page, R. L., Bustamante, C. D., & Rivas, M. A. (2020). FasTag: Automatic text classification of unstructured medical narratives. PLoS ONE, 15(6 June). https://doi.org/10.1371/journal.pone.0234647

153. Venkatesh, B., & Anuradha, J. (2019). A review of Feature Selection and its methods. Cybernetics and Information Technologies, 19(1), 3–26. https://doi.org/10.2478/CAIT-2019-0001

154. Wang, H., Tian, K., Wu, Z., & Wang, L. (2021). A short text classification method based on convolutional neural network and semantic extension. International Journal of Computational Intelligence Systems, 14(1). https://doi.org/10.2991/ijcis.d.201207.001

155. Wang, S., Jiang, L., & Li, C. (2015). Adapting naive Bayes tree for text classification. Knowledge and Information Systems, 44(1), 77–89. https://doi.org/10.1007/S10115-014-0746-Y/METRICS

156. Wang, Y., Sohn, S., Liu, S., Shen, F., Wang, L., Atkinson, E. J., Amin, S., & Liu, H. (2019). A clinical text classification paradigm using weak supervision and deep representation. BMC Medical Informatics and Decision Making, 19(1). https://doi.org/10.1186/s12911-018-0723-6

157. Wang, Y., Zhou, Z., Jin, S., Liu, D., & Lu, M. (2017). Comparisons and Selections of Features and Classifiers for Short Text Classification. IOP Conference Series: Materials Science and Engineering, 261(1). https://doi.org/10.1088/1757-899X/261/1/012018

158. Wongso, R., Luwinda, F. A., Trisnajaya, B. C., Rusli, O., & Rudy. (2017). News Article Text Classification in Indonesian Language. Procedia Computer Science, 116. https://doi.org/10.1016/j.procs.2017.10.039

159. Wu, H., Li, D., & Cheng, M. (2019). Chinese text classification based on character-level CNN and SVM. International Journal of Intelligent Information and Database Systems, 12(3). https://doi.org/10.1504/IJIIDS.2019.102940

160. Wu, H., Liu, Y., & Wang, J. (2020). Review of Text Classification Methods on Deep Learning. Materials & Continua CMC, 63(3), 1309–1321. https://doi.org/10.32604/cmc.2020.010172

161. Xu, S., Li, Y., & Wang, Z. (2017). Bayesian multinomial naïve bayes classifier to text classification. Lecture Notes in Electrical Engineering, 448, 347–352. https://doi.org/10.1007/978-981-10-5041-1_57/COVER

162. Yan, D., Li, K., Gu, S., & Yang, L. (2020). Network-Based Bag-of-Words Model for Text Classification. IEEE Access, 8. https://doi.org/10.1109/ACCESS.2020.2991074

163. Yan, Y., Wang, Y., Gao, W. C., Zhang, B. W., Yang, C., & Yin, X. C. (2018). LSTM 2 : Multi-Label Ranking for Document Classification. Neural Processing Letters, 47(1), 117–138. https://doi.org/10.1007/S11063-017-9636-0/METRICS

164. Yang, X.-S. (2009). Firefly Algorithms for Multimodal Optimization. Lecture Notes in Computer Sciences, 5792, 169–178.

165. Yao, L., Mao, C., & Luo, Y. (2019). Graph convolutional networks for text classification. 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019. https://doi.org/10.4000/books.aaccademia.4577

166. Yao, T., Zhai, Z., & Gao, B. (2020). Text Classification Model Based on fastText. Proceedings of 2020 IEEE International Conference on Artificial Intelligence and Information Systems, ICAIIS 2020. https://doi.org/10.1109/ICAIIS49377.2020.9194939

167. Yelp Dataset. (n.d.). Retrieved September 18, 2023, from https://www.yelp.com/dataset

168. Yoshioka, K., & Dozono, H. (2018). The classification of the documents based on Word2Vec and 2-layer self organizing maps. International Journal of Machine Learning and Computing, 8(3). https://doi.org/10.18178/ijmlc.2018.8.3.695

169. Yu, M., Huang, Q., Qin, H., Scheele, C., & Yang, C. (2019). Deep learning for real-time social media text classification for situation awareness – using Hurricanes Sandy, Harvey, and Irma as case studies. Https://Doi.Org/10.1080/17538947.2019.1574316, 12(11), 1230–1247. https://doi.org/10.1080/17538947.2019.1574316

170. Zaki, T., Es-saady, Y., Mammass, D., Ennaji, A., & Nicolas, S. (2014). A hybrid method N-Grams-TFIDF with radial basis for indexing and classification of Arabic documents. International Journal of Software Engineering and Its Applications, 8(2). https://doi.org/10.14257/ijseia.2014.8.2.13

171. Zhai, Y., Song, W., Liu, X., Liu, L., & Zhao, X. (2019). A Chi-Square Statistics Based Feature Selection Method in Text Classification. Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS, 2018-November. https://doi.org/10.1109/ICSESS.2018.8663882

172. Zhang, X., Zhao, J., & Lecun, Y. (n.d.). Character-level Convolutional Networks for Text Classification *.

173. Zhang, Y., Marshall, I., & Wallace, B. C. (2016). Rationale-augmented convolutional neural networks for text classification. EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings. https://doi.org/10.18653/v1/d16-1076

174. Zheng, J., & Zheng, L. (2019). A Hybrid Bidirectional Recurrent Convolutional Neural Network Attention-Based Model for Text Classification. IEEE Access, 7. https://doi.org/10.1109/ACCESS.2019.2932619

175. Zhou, C., Sun, C., Liu, Z., & Lau, F. C. M. (2015). A C-LSTM Neural Network for Text Classification.

176. Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., & Xu, B. (2016). Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. COLING 2016 - 26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers.

177. Zulqarnain, M., Ghazali, R., Ghouse, M. G., & Mushtaq, M. F. (2019). Efficient processing of GRU based on word embedding for text classification. International Journal on Informatics Visualization, 3(4). https://doi.org/10.30630/joiv.3.4.289

