RES MILITARIS
REVUE EUROPEENNE D ETUDES EUROPEAN JOURNAL OF MILITARY STUDIES
Social Science Journal

Literature Review: Challenges In Creating A Generic Model For Text Classification In Multiple Languages

By

Rachana Sinha
School of Computer Applications, Babu Banarasi Das University, Lucknow, India
Email: rachanasinhamgp@bbdu.ac.in

Dr. Reena Srivastava
School of Computer Applications, Babu Banarasi Das University, Lucknow, India
Email: dean.soca@bbdu.ac.in

Abstract

Since its inception in 1960, text classification has had a long history. Since then, Text classification has benefited much from the study. Some studies have modified the original tools, while others have incorporated innovative ideas in one or more steps of the text classification approach, but no general model has yet been proposed that can efficiently classify the text in multiple languages. This research examines the literature in twenty languages to determine what technical challenges exist that prevent the creation of a generic model. It also seeks to determine whether it is possible to develop a model that can classify text in many languages. Our findings show that The phases of representation and pre-processing are found to bring particular challenges for each language. Depending on the availability of the data and the precise classification objective, academics classify text published in languages other than English using essentially the same basic methodology, tools, and other components. There is no single Generic model that is acceptable across all languages, despite the adoption of comparable models in text classification across several languages. The questions posed as a research challenge for this study are addressed in this paper. Our findings will aid scholars in comprehending the problems encountered at various stages of the text classification process. It will also assist them in thinking of innovative ways such as reducing existing bottlenecks and opening up new study fields.

Keywords: generic model, literature review, multiple languages, pre-processing, text classification

Introduction

The classification of unstructured texts into pre-set categories has become a critical need of the hour in light of the current reality of massive online textual data being generated every second. Text classification, on the other hand, dates from the 1960s. Since the 1980s, several scholars have been working on text categorization, but no universal model that can categorize all types of unstructured text has been developed. The goal of this review study is to identify a likely solution to the two questions below:

Q1. What are the technological challenges during the different phases of the categorization procedure that limit the building of a generic model?
Q2. What is the best way to get a generic model that can classify, if not all, but at least the text of two or more languages?

To address the above two concerns, one must first comprehend the basic processes of the text classification procedure as well as the expectations from a generic classification model. The human technique of classifying a collection of given text into some pre-defined categories would most likely start with understanding the categories, reading the text, applying their knowledge to comprehend the context, and inferring the relationship between the context and the pre-defined categories and as a result, the text might be categorized into any of the available categories.

The machining method of classifying a set of text, on the other hand, starts with reading the text and pre-processing it, eliminating the less important tokens while maintaining the most significant ones in a succinct form, quantifying the text for numerical transformation, and then applying mathematical methods to categorize it into a given category. A generic model for text classification can be defined as a model that can read any type of text (text in more than one language; The scope of this study does not extend to any other kind of text), and represent the text in some intermediate numeric format, and perform some computation to determine which category the given text belongs to.

Our analysis began with a survey of prior literature reviews conducted by different researchers in various languages. These evaluations aided us in determining the distinctiveness of our paper. The following is a list of prior literature reviews. After this discussion different relevant sections have been used to present the review of literature linked with other steps of the text classification in different languages.

A. Previous Literary Analyses

One of the prior literature reviews, [1] provided a thorough overview of the text classification process, including its phases and methodologies while other researchers [2] looked at opinion mining and sentiment classification in Hindi, Russian, and Chinese, and discovered that most non-English research followed the approaches employed in English, with only limited use of language-specific aspects such as morphological differences. Cross-domain research has received less attention. Some of the researchers [3] have given a year-by-year comparison of different approaches in each step of the classification process. They have provided their thoughts on the circumstances in which one solution outperforms others on a given application challenge.

In a study [4] four writers each defined automatic text classification, which has been expressed in mathematical notation and graphical form. The researchers have prioritized the research of external knowledge's role and impacts on automatic text classification. Some of the researchers [5] have conducted a literature review of text classification papers published between 1997 and 2012 and discovered that 86 percent of the publications employed machine learning-based methods for text categorization. They discovered that SVM and KNN algorithms were employed in 65 percent of the studies. They also looked at various articles on multilingual text categorization and discovered that LSI and SVM are complementary and that a hybrid model is needed to overcome both models' limitations.

A thorough literature review [6] was presented that outlined the major features of the various Text Classification approaches and methods used to classify Arabic text. The researchers complained about the absence of standardized corpora for Arabic text, noting that the majority of the work had been focused on Modern Arabic text, with Colloquial Arabic receiving relatively little attention, while some of the reviewers [7] discussed the advantages, disadvantages, and current research trends in Artificial Intelligence. They have underlined the importance of understanding the data's nature before mining it. They also recommended looking into developing simpler algorithms, a superior method for integrating domain

knowledge, multilingual text refining, subjectivity detection, and contrastive viewpoint summarization. A prior review paper [8] analyzed existing text classification algorithms, highlighting the key constraints of each component of the text classification process. One of the studies [9] investigated contemporary neural network trends for classifying Arabic text. They provided evidence for the use of deep learning models to enhance Arabic text classification research. Based on difficulties and research correlation of previously proposed text classification phases and associated challenges, researchers [10] have reported trends, gaps, and research patterns of text classification techniques. They discussed nine various sorts of research objectives, as well as diverse datasets, text language usage, proposed technique focus areas, and the gap in the last twelve years while some researchers [11] investigated text classification pre-processing, feature extraction, alternative methods, and approaches, as well as evaluating performance matrices for improved evaluation.

To our knowledge, no comprehensive systematic literature review has analyzed the reasons for using different text classification techniques or a different combination of methods in all steps of text classification, even though many researchers and scholars have reviewed the previous literature descriptively and systematically with different points of view. This review paper examines the complete text classification technique from the perspective of understanding the role and impact of each phase on the overall text classification operation, to determine the possibility of a generic model. To uncover the answers to the study questions, a more in-depth analysis was conducted while reviewing the literature linked to each step of the text categorization technique. In addition to these literature evaluations, we looked at work done by researchers in about twenty languages, such as(Arabic, Chinese, Croatian, English, Finnish, French, German, Hindi, Indonesian, Japanese, Korean, Marathi, Persian, Polish, Portuguese, Russian, Sanskrit, Spanish, Turkish, Uzbek), to figure out why no generic model exists and what technological obstacles it faces.

The remainder of the paper is laid out as follows. The investigation and analysis of the pre-processing step are discussed in Section II of this paper. The combined investigation and analysis of text representation, feature selection, and feature extraction procedures is presented in Section III. The review and analysis of text classification models are presented in Section IV. Section V discusses the conclusion as well as the answers to the research issues under discussion. In section VI, the study concludes with a list of references.

Pre-Processing

Preparing text data for a classification system by using a technique called text pre-processing. Noise in text data includes varied text cases, punctuation, and other elements. A few of the pre-processing methods used are tokenization, stop word filtering, parts-of-speech (POS) tagging, word sense disambiguation, grammatical parsing and cleaning, lemmatization, stemming, text summarization, document indexing, and TF-IDF. In this section, we have provided our study of several related pieces of literature and attempted to identify the challenges that researchers experience throughout the pre-processing step. We investigated not just pre-processing of English language text, but also pre-processing of text in other foreign languages. We also looked into several text pre-processing approaches for Indian languages. Although this is not a full collection of articles, we have attempted to include the most significant ones. In the following section, we have presented the studies that shed light on the pre-processing step's consequences.

B. Effects of Pre-Processing

Pre-processing is an important part of text categorization since it cleans up the data by removing noise and irrelevant information. In this area, we've included various papers that address the impact of pre-processing on text classification. The following papers were chosen to represent this section because they demonstrate how text classification is impacted by pre-processing in a variety of languages, including German, English, Turkish, Arabic, Chinese, and Spanish.

Investigators [12] created a multi-label text classification system to sort free-text medical documents written in German into pre-defined categories. They compared the performance of Naive Bayes, k-NN, SVM, and J48 classification models. They also ran additional tests to see if pre-processing affected the results. Their findings revealed that pre-processing enhanced performance, with J48 achieving the highest outcomes. Arabic is a member of the Semitic family of languages. Researchers [13] explored the effect of Arabic text pre-processing on Arabic text classification. They concluded that reducing the number of features lowered classifier complexity and space needs while also saving time.

The impact of pre-processing on English and Turkish text categorization in terms of accuracy, text domain, text language, and dimension reduction was investigated in the paper [14]. The researchers compared all conceivable combinations of widely used pre-processing activities on two domains: email and news. They discovered that selecting the right combination of pre-processing activities can result in significant performance improvements. Chinese stop words contain very valuable information, so cannot be directly removed just like English stop words.

[15] studied how text pre-processing affected text categorization using machine learning methods. The researcher employed tokenization, stop word removal, and stemming as pre-processing methods, and then compared Chi-square and TF-IDF with cosine similarity scores for feature extraction. The findings of the experiment revealed that text pre-processing had an impact on feature extraction approaches that improve English text classification performance, especially for modest threshold values. Tw-stAR is a multi-label classification system developed by researchers [16] that recognizes the emotions expressed in Arabic, English, and Spanish twits. They discovered that stemming, lemmatization, and emoji labeling were the most successful tasks for emotion MLC, emphasizing the importance of pre-processing in emotion MLC. The effects of various pre-processing techniques on text classification in Spanish and English across short and lengthy documents were examined in [17]. They recommended using lemmatization in the short dataset and stop words in the large dataset based on the positive outcomes of their experiments. In comparison to the count vectorizer, which produced the greatest results for short text, TF/IDF has an impact on classifier performance by raising the f-measure for long text. They demonstrated that the best number of characteristics for long texts is 500, compared to 50 to 100 for short texts.

The researchers [18] have discussed the strategies they employed in their Russian chatbot assistance for structuring the employee support system. For pre-processing the Russian language text, NLTK, Pymporphy2, SpaCy, gensim, and MyStem were used to investigate approaches such as tokenization, elimination of stop words, and reduction to the basic form. They also contemplated using the Deeppavlov framework to recognize named entities. While another project on identifying abusive text in Russian [19] concluded that by selecting the appropriate preprocessing techniques and language-specific feature selection, it is possible to achieve state-of-the-art performance on par with the best-performing English language models even using a simple SVM model.

After reading multiple articles about several languages, we discovered that pre-processing has a significant impact on the text categorization approach. In some situations, it improved the classifier's performance and throughput, but in others, no significant changes were seen. Aside from that, we learned that pre-processing comprises techniques like tokenization, stop word removal, POS tagging, stemming, and lemmatization. Our research found that different combinations of various pre-processing techniques had the most positive impact on the categorization method, though it was unclear which specific methods should be combined to achieve the desired result, and why a particular combination of methods was chosen in research papers. There was no mention of it in any of the articles. In the next section, we have presented some of the works that talk about Tokenization.

C. Tokenization

A textual passage can be divided into smaller components called tokens through the process of tokenization. Traditionally tokenization was done on a word level [20]. Recently more granular decomposition has been applied like character n-gram, sub-words, and even byte representation of text.[21]. Researchers [22] have emphasized that tokenization can be regarded as the most important operation among other pre-processing methods that are language-dependent. Earlier rule-based tokenization was carried out by separating tokens along white spaces, punctuations, and contractions. Mosses [23] and Spacy [24] are rule-based NLP toolkits. Tokenizers decompose the input text data and create a vocabulary of terms that are used to generate index-based numerical representation. These representations are sometimes very large having out-of-vocabulary (OOV) words from which a model cannot extract meaningful information. To reduce these OOV words, some researchers have proposed character level segmentation which works well in some languages but was not relevant in sequence modeling [25]. Byte pair Encoding BPE was an important development in tokenization [26].

Earlier it was used as a data compression algorithm but later it was used for sub-word segmentation [27]. Byte level BPE [28] was applied to raw bytes. Word piece Tokenizer[29] was developed for Japanese text segmentation problems similar to word piece.UnigramLM was proposed [30] which worked in the opposite direction of Word Piece Tokenizer.

Many of the monolingual tokenization techniques rely on extra-linguistic preparation. Before utilizing the WordPiece algorithm, Arabic tried pre-segmentation techniques, Japanese used a pre-built morphological parser whose tokens were then converted into characters, and Korean included bi-directional conditioning. It can be concluded that text segmentation is a crucial component of the Text Classification process with high linguistic relevance and significant implications for every step of a classifier's pipeline because of its inextricable links to embedding formation. The language dependency of the tokenizer is the probable reason for the non-existence of a generic model for tokenization. We've listed some of the works that discuss the Stop word removal strategy in the text categorization method in the next section.

D. Stop Word Removal

In any natural language, the most common words that do not add much value to the meaning of the document while analyzing the given text document, are called Stop words. Domain-specific words also contribute to stop words of any language. These stop words are removed from the document before further processing. In this subsection, we have presented some of the papers that discuss the novel methods of removing stop words in languages such as Arabic, Chinese, English, French, and Sanskrit. For the Arabic language researchers suggested a stop word removal algorithm using a Finite State Machine [31] that achieved 98 percent accuracy. Chinese language researchers proposed a probabilistic automatic aggregated

methodology-based algorithm [32] however, they did not test it for other languages. They utilized a mix of domain and dispersed generic terms.

For the Spanish language, a new method for the selective eradication of stop words in the term candidate was put out [33]. Some of the choices for the term had stop words included in them. With the use of a predefined stop word list, these words are often removed when stop words are removed. The researchers advised simply removing the stop word portion of the word rather than the complete word because the remaining portion may contain a significant candidate phrase. The researchers [34] proposed that for English and French, a smaller stop word list of 9 words performed similarly to a bigger list of 571 terms. In Hindi and Persian, a bigger list improved the outcome. Some of the experimenters [35] designed a stop word removal algorithm based on a hybrid dictionary approach and implemented it in Sanskrit with 98 percent accuracy. Investigators [36]have offered a systematic review of various stop word removal strategies. They discussed the classical approach, Zipf's law method, the mutual information method, and the term-based random sampling method.

According to our investigation based on the above-mentioned literature, each language has its own set of stop words. Even within the same language, this set of stop words might vary depending on the context of the document. As a result, researchers devised novel stop word removal strategies in a variety of languages, which yielded positive results. Still, there is a lack of language-independent universal techniques for stop word removal. Apart from stop word removal, we discovered that stemming and lemmatization were the most commonly used stages in pre-processing text documents. The next subsection looks at some of the papers that examined stemming.

E. Stemming

Stemming is the process of condensing words with inflected forms to their word stem and root forms. It is one of the most essential and often used methods for assisting in-text normalization. This section contains papers on the stemming operation in text categorization for various languages such as Portuguese, English, Indonesian, Marathi, French, Spanish, Uzbek, Japanese, and Arabic.

Spanish has a sophisticated morphology. The investigator [37] suggested a straightforward yet effective algorithm. They proposed 300 stemming and reduction rules that were paired with a dictionary search. By using various examples, they showed that their experiment worked. On the other hand, some researchers used the SUM paradigm on the Reuter dataset and the European Portuguese language dataset in their experiment[38]. Using all of the terms in the English dataset and stemming them improved the classifier.

The goal of the Stemming method is to locate a word's morphological root. In the paper [39], researchers have described various types of languages, ranging from isolating language (no morphology) to polysynthetic language (complex words with more morphemes). There is also agglutinative language, which has words that can be easily separated into morphemes, and fusional language, which has words that are not identifiable morphemes. Because different languages have distinct morphological rules, there is no such thing as a perfect stemmer that can reliably extract the stems of each phrase regardless of its properties.

Words in the Indonesian language have prefixes, suffixes, infixes, and confixes, making it difficult to match related words. Researchers [40]offered a revision to the Nazief and Adriani method and attained a 95% accuracy rate. Unlike other Indian languages, Marathi words have a consistent structure. Plain suffixes, connect word suffixes, and complicated suffixes are the three forms of suffixes in Marathi. An unsupervised stemmer based on the n-gram splitting

technique has been proposed [41]. They recorded maximum accuracy of 82.5 percent. The utility of stemming in an application is determined by the nature of the input document, the vocabulary, and the application's aim. Researchers [42]compared all three types of stemming algorithms and discovered that some stemming algorithms performed better in one area while others performed better in another. They concluded that no perfect stemmer had yet been designed to meet all of the parameters.

A new approach was introduced by the experimenters [43] by truncating each word to its beginning letter. The new ultrastemming approach significantly decreased the volume of representation while keeping the substance of the summary, and it also enhanced the system's performance. They validated their findings using corpora from three languages: English, French, and Spanish. While researchers [44] discovered that Uzbek is a morphologically rich agglutinative language. Since most stemmers are language-dependent (English being the most popular), they proposed a stemmer based on Lovins Stemmer for the Uzbek language. To meet the needs of the Uzbek language, they improved the characteristics of Lovins Stemmer by adding the ability to remove prefixes as well (Lovins Stemmer is a suffix-removing model).

Stemming and lemmatization are adventitious in the areas of Arabic IR and NLP applications [45]. The effectiveness of information retrieval is significantly increased when Arabic roots are used as indexing phrases. It seeks to lower the size of data to increase transmission speed and decrease the size needed for storage. In the case of Arabic, overly semantic classification affects roots, whereas underly semantic classification affects stems. Lemmas are a static lexicon at a specific point in time that might not share all the words' grammatical characteristics.

Researchers [46] conducted a literature assessment of all Japanese pre-processing methods and tools and discovered that morphological analysis of the Japanese language is challenging owing to the Japanese grammar system's peculiarity. As a result, they advised that to complete the Japanese pre-processing phase, a mix of Japanese NLP technologies should be used.

Our research reveals that each language has its morphology and grammar, which limits the construction of a universal stemmer that can handle numerous languages. The lack of a list of root words and their probable inflections further hinders the construction of a stemmer of this type. A thorough investigation of the stemming process in both English and other languages [47] contends that because no language perfectly adheres to a set of deterministic principles, it is challenging to create a perfect stemmer that can precisely determine the stems of any phrase regardless of its properties. Recall and Precision have been used to gauge the stemmer's effectiveness, however, these measures are susceptible to outside influences. The needs of the user affect the choice of the stemmer as well. For example, depending on the space available in the device, a robust stemmer or a light stemmer might be employed. The application's goal and the collection's document length have a direct bearing on the outcomes. The next subsection talks about lemmatization and POS tagging.

F. Lemmatization and POS Tagging

Lemmatization is the process of organizing a word's inflected forms into a single unit for analysis using the word's lemma, or dictionary form. It considers the context in which the word is being used hence giving a grammatical valid word as a lemma. When a word in a text is marked up as belonging to a specific part of speech based on both its meaning and context, this process is known as POS tagging. This subsection presents a few papers that discuss lemmatization and POS tagging in different languages such as English, Hindi, Arabic, and French.

By presenting a model based on linguistic pre-processing supported lemmatization, researchers [48] showed that their approach fared well on the well-known Reuter3 dataset. Academics [49] presented a novel method for addressing lemmatization as a classification task in machine learning. To transform the input word into an output lemma, the researcher computed an SES (Shortest Edit Script) between the reverse input and output strings. This method performed well and had a high level of language independence.

Hindi is a language with a lot of inflections. Based on the core notion of space and time optimization, investigators [50] presented a Hindi lemmatizer. After processing a corpus of 40000 phrases with 75 lakh words, they came up with a list of 124 suffixes. They've also devised a set of 124 rules for removing suffixes from root words and, if necessary, adding a character or 'matra' to the root word. They claimed that their technique was accurate to the tune of 89.08 percent. Experimenters [51] demonstrated that by employing POS features to classify Amazon product review data, the multiclass classification technique has a greater classification accuracy.

Some researchers [52] described the use of the TXM platform (http://textometrie.org) to lemmatize Medieval French literature. Their project compiled an open morphological lexicon of Medieval French using available lexical resources. A human expert verified and corrected the lemmas at the end of the process. The authors [53] used a corpus of 2.2 million tokens that were annotated and validated using machine learning and a lemmatization dictionary to train their machine learning model for Arabic text. While another Arabic lemmatizer that gives a single lemma to each word in an Arabic sentence while taking into account the word context was proposed by [54]. Two modules make up the proposed system. A tagged corpus of roughly 500,000 words was used to validate this method's validity.

Pre-processing has been identified as a key step in text classification, with implications for the total method, according to our findings. Languages have their morphology and grammar, as well as their own set of limitations. As a result, researchers have experimented with various combinations of pre-processing procedures to achieve the best outcomes. According to our findings, this is the probable reason for not having any generic model for pre-processing of text in multiple languages. The following section of our paper discusses text representation approaches and includes literature evaluations on the subject.

Text Representation And Dimensionality Reduction

The objective of text representation is to find acceptable phrases to convert a document into numerical vectors that capture the same semantic information. The ability to infer vectors from text, whether at the character, word, phrase, or document level, has led to the development of numerous strategies and methods over time. Discrete Text Representation and Distributed Text representation are two types of Text Representation. Discrete Representation includes One-Hot Encoding, Bag-of-words, Count Vectorizer, and TF-IDF, whereas Distributed Text Representation includes Co-Occurrence Matrix, Word2Vec such as CBOW and Skip-gram, GloVe, Fast Text, and BERT. Dimensionality Reduction is done through feature selection or feature extraction methods. These techniques help in the compressed representation of the document. We have offered an overview and analysis of previous work in the field of text representation in this part. We aimed to include research projects that used a novel method of text representation in any of the languages such as English, German, Croatian, Chinese, Japanese, Russian, Arabic, French, Korean, and Finnish. The pieces of literature studied in this section have been placed into four sub-sections Discrete Text Representation Distributed Text Representation and Some Novel Approaches and Dimensionality Reduction.

A. Discrete Text Representation

In the case of text classification of German language texts using SVM, [55]determined that term frequency changes have a greater impact on SVM performance. On a Croatian-English parallel dataset, [56]compared the n-gram and morphological normalization methods. Their findings revealed that n-grams improved classifier performance and could be used instead of morphological normalization, although they were computationally expensive. Another study talked about a bag-of-visual-words representation that is used to classify scenes. This representation is similar to the bag-of-words representation for text documents. Researchers [57] revealed that it can give an empirical foundation for constructing visual-word representations that are likely to provide higher classification performance.

Some investigators [58] found that adding sentences to unigrams improved the categorization outcome for English patents substantially. The results were tested for generalizability to French and German, where the German language did not benefit. Whereas to understand the impact on the accuracy of the Support vector machine for sentiment analysis in Spanish, researchers [59] did a thorough investigation of various text transformations, tokenizers, and token weighting techniques. They concluded that in their circumstance, q-gram tokenizers performed better than n-word tokenizers.

B. Distributed Text Representation

Many NLP tasks use distributed word representations or word vectors that have been pre-trained on vast amounts of data. Authors [60] used a dataset that included Wikipedia and Common Crawl to train the Skip-gram and CBOW word vector models. They also developed a quick language identifier that recognizes 176 different languages. While researchers [61]evaluated the impact of employing multi-words for representation on text categorization performance. They showed that subtopic representation beat general idea representation in multi-word representation. The linear kernel of the SVM outperformed the nonlinear kernel in classifying the Reuter21578 dataset. Some of the researchers [62] introduced a classifier that used Wikipedia to encode articles as vectors of concept weights, and they evaluated its usefulness for identifying biomedical texts published in any language while it was exclusively trained on English data.

Graph neural networks are useful tools for analyzing graph-structured data. Authors [63] used a statistical word co-occurrence network to represent text material. The results were encouraging, as the proposed architecture was competitive with the existing ones. On the other hand, some of the researchers [64]developed a new way of describing and recognizing confounding variables in text categorization. They claimed that their model can learn textual representations that are unaffected by confounding variables.

In Chinese, English, Japanese, and Korean, researchers [65] have investigated the use of various encoding techniques for both deep learning and linear models for text classification. They concluded that the best encoding strategy for convolutional neural networks was byte-level one-hot encoding.

Authors [66] presented a novel Finnish BERT model that was trained from the ground up and compared it to M-BERT on published datasets for POS tagging, NER, and dependency parsing, as well as a variety of text classification tasks. They claimed that their unique BERT model outperformed all other proposed models, including multilingual models. Whereas researchers [67] introduced an adaptive neural network technique based on the generalized Hebbian Algorithm to extract the first principal component of a French corpus of 90 web pages to reduce the dimension of the corpus. This algorithm only needed one vector at a time, ran quickly, and delivered accurate results.

C. Some Novel Approaches

To represent the text document, the researchers [68] suggested a unique Tensor Space Model (TSM) based on algebraic high-order tensors at the character level. TSM outperformed VSM on 20 Virtanen et al.'s Newsgroup datasets, according to the experimental results. To define the class to which the document belongs, investigators looked for textual patterning [69]. Here feature vectors and tree kernels were produced by the tree-like text representation. These kernels were then utilized as a model selection technique in supervised learning based on cross-validation. A feature selection and word weighting strategy based on ontologies was examined by certain authors. [70]. They compared the BOW to the TFIDF method, which is based on domain knowledge. The outcome on a small dataset in the IT area indicated a 10.93% improvement in VSM performance. Representing Arabic Text semantically using Rich Semantic Graph (RSG) is one of the recent techniques that facilitate the process of manipulating the Arabic Language. The study [71] by the authors is a component of continuing research to produce an abstractive summary for a single Arabic input document. The Text to RSG representation is one of three modules that uses a domain Ontology to abstract Arabic text summarization.

D. Dimensionality Reduction

For Arabic language publications, experimenters used an SVM-based text categorization algorithm [72]. The F-value their model generated was 88.11. Researchers suggested using a Unified manifold learning framework for unsupervised and semi-supervised dimension reduction. [73]. To map the new data points, they used a straightforward yet successful linear regression algorithm. Their method could leverage several structures from labeled and unlabeled data as well as label information from labeled data. The CHI Square methodology was used as a feature selection method, and attribute overlap minimization and outlier elimination were used as dimensionality reduction strategies, respectively, for text categorization by authors [74]. They reported considerable gains in prediction precision, tree size, and space. Additionally, their model performed well, especially in huge vector spaces.

Some academics suggested a new dimension reduction method based on hierarchical agglomerative word clustering [75]. A Co-operative Coevolutionary genetic algorithm was used to optimize the cluster weight of these clusters. For text pre-processing, they employed TF-IDF and Conf weight.

In CNN, some of the researchers implemented an active learning strategy [76]. They created the model to swiftly discriminate between different word embeddings for a certain job. The novel approach outperformed the baseline AL approach on both a sentence and document level, according to the findings.

On the other hand, some suggested an improved feature selection method that employed word embedding to calculate the most comparable terms to the current vocabulary determined by the IG algorithm and extend the lexicon with these words while adhering to specific rules [77]. In the Sogou Chinese text corpus and the Fudan Chinese text classification dataset, their model performed well. Some researchers[78] compared two methods for extracting characteristics for subject categorization of Polish Wikipedia articles with 34 subject areas: BOW and Word embedding techniques. Their findings revealed that a feature selection strategy based on word embedding outperformed typical NLP features, but that it required a suitably big training dataset.

Fisher discriminant analysis was used by researchers [79] for dimensionality reduction in Arabic text categorization, and they found that it is 84.4 percent accurate. Some authors examined several fuzzy and support vector machine-based text document classification

systems' feature reduction and dimensionality reduction parameters [80]. Their comparison studies' results demonstrated that feature extraction and the support vector machine were the most popular options among the current systems. Compared to all other methods now in use, their usage percentages are 43.3 percent and 26.08 percent, respectively. Neural networks are routinely employed to learn how to represent text. Authors [81] designed LSA to take the representation a step further by performing dimensionality reduction on a Document word matrix. The focus switched from feature engineering to learning with probabilistic models like PLSA and LDA.

As the text input is transformed into a matrix of numbers, data representation is a key stage in text classification. Our findings show that scholars from many languages have experimented with various text representation models to improve the overall outcome and performance. Although some scholars applied the same representation approach to multiple language texts, not every work benefited. There is no explanation for why the model failed to get the desired outcomes for different languages. It can be deduced that a generic model for text representation for several languages is urgently needed and that more research is required to establish a general text representation model for various languages.

Classification Models

The selection of a classification model has been the focus of the majority of research in the text categorization process. This field has benefited from the contributions of several researchers. Some have refined existing models, while others have presented a completely new one. Others have merged ideas from several disciplines to build entirely new models. Although we attempted to review some of the work in this section, it was impossible to cover all of the work done on this subject. The data is no longer language-dependent after the text representation stage, and the model for English language text can be applied to other languages as well. Text classification models are divided into two types: n-gram models like Logistic Regression, Simple Multi-Layer Perceptron, Support Vector Machine, and sequence models like CNN, RNN, and its variants. This section has been divided into four sub-sections such as n-gram models, Sequence Models, hybrid Models, and Novel models to present the literature review in a systematic order and for ease of understanding.

A. N-gram Models

Authors [82] achieved 89 percent accuracy for the linear SVM classifier while working on a corpus of Polish press news without pre-processing or normalization. This led to two conclusions: language system complexity had no significant impact on the TC process, and reducing vector dimensions improved TC performance for Polish but had no effect on TC effectiveness.

B. Sequence Models

Earlier some researchers [83] proposed an artificial neural network with SVD for the Arabic corpus that achieved a score of 88.33 percent, higher than the simple ANN's score of 85.75 percent. They used the term-weighting technique to represent each Arabic document. On the other hand, researchers [84] improved French Text classification with the use of recurrent convolutional neural networks. In their experiment, a recurrent structure gathered contextual information, and a convolutional neural network generated a text representation. The model outperformed CNN and Recursive NN. The text region embedding +pooling framework was studied by[85], who found that LSTM combined with convolutional layers produced the best results.

The experimenters [86] proposed a text classification architecture using a word-level deep CNN architecture of minimal complexity that can effectively represent a long-distance relationship in text. Authors [87]

examined text representation label embeddings and suggested a label embedding attentive model. It embeds words and labels in the same joint space and assesses the word-label compatibility to attend to document representation that incorporated position invariance into RNN, overcoming the disadvantages of both RNN (not excellent at extracting keywords) and CNN (not good at extracting keywords) (long term dependencies). By incorporating topic information into bidirectional LSTM, researchers [88]demonstrated a new approach for searching a document collection for arguments pertinent to a particular topic. For the first time, researchers [89] fine-tuned BERT to improve text classification baselines. Authors[90] used a combination model (LSTM+CNN) to classify French reviews and reported 93.7 percent accuracy.

C. Hybrid Models

Some researchers [91] combined limited one pass clustering with KNN to construct an incremental classification model. The training data were compressed during the constrained one-pass clustering, which also revealed the complex distribution. They demonstrated through experiments that their model performed better than KNN, Naive Byes, and Support Vector Machines and that it has excellent adaptability and scalability in real-world applications.

Authors [92]suggested an LSTM-CNN hybrid model that significantly increased text categorization accuracy. [93] submitted their CNN-based model in July 2019 that simultaneously learns segmentation boundaries and stems from the Myanmar language. Their text representation model used character and syllable levels. They concluded that the pre-trained embedding significantly affects performance. To identify sarcasm in Spanish text, researchers [94] created a model based on the Transformer model's encoder component. Transformer encoders can model the complicated long-range relationships between text phrases without the use of convolutional or recurrent layers. Considering that the results were obtained without performing lengthy model hyperparameter experiments, they were quite encouraging. The researchers introduced the TGNet (Temporal Convolutional Network), a hybrid neural network that uses GRU for context modeling and TCN to identify the link between hidden features across temporal scales[95].

The multiclass imbalance is still a concern in real-world data mining and machine learning. Researchers used WEKA to create a hybrid classifier[96]. Hybrid ensemble classifiers using Random Forest performed significantly better than single classifiers. A novel hybrid Convolutional Genetic model for Arabic Text was developed by [97]to achieve a high classification accuracy by overcoming the parameter setting issue. Whereas [98] employed pre-trained versions of the BETO, RoBERTa-Large, and RoBERTa-Large +CNN models to categorize fake news and legitimate news in Spanish. For estimating the three models' outcomes, he employed the ensemble approach. In comparison to the other two, RoBERTa-Large+CNN provided the best F1 score (0.6860), which was 0.806 points lower. To achieve the optimum outcome, the researcher advised further adjusting the parameter and attempting more integrated learning techniques. Some researchers [99] pre-trained m-BERT, BETO, and XLM-RoBERT on a corpus of real-world cancer clinical cases before fine-tuning these models for three different tasks for clinical coding in Spanish. In all three challenges, the domain-specific model fared better than the original general domain model. The ensemble method's aggregate result significantly outperformed the prior performance by 11.6 percent, 10.3 percent, and 4.4 percent, respectively.

D. Novel Models

In order to improve the effectiveness of categorization and unsupervised text classification, researchers [100]developed the OTTO framework, which used text mining to learn the target ontology from a text source. Some authors [101] classified Chinese and Malay sentences and evaluated the effectiveness of their model against that of English. They then used novelty mining to find the sentences that included new information. The TREC Novelty Track data was used in their experiment, and they concluded that text classification is crucial for novelty mining and that their model outperformed Chinese.

RMDL is a novel method developed by investigators [102] that can accept text, video, pictures, and symbolic input. RMDL produced better results, according to the findings. For the categorization of Arabic text, other researchers [103] suggested a deep Autoencoder-based representation. Their approach combined implicit and explicit semantics, allowing for the study of document semantics. The results gained demonstrated the efficiency of their suggested strategy in comparison to cutting-edge ones. Capsule networks have been suggested by certain academics [104] as a solution to the text categorization issue. They evaluated the proposed model to CNNs using seven standard benchmark datasets. For text classification, capsule networks are helpful and offer a cheap substitute for dynamic routing.

Another investigator suggested a promising peer learning model for distinguishing between actual and expected labels that could reduce mistakes [105]. While another [106] provided a hyper-heuristic strategy for tweaking the hyperparameters of recursive and partition trees. The proposed method was tested on 30 different datasets. HEARpart outperformed WEKA's J48 method in terms of error rate, F-measure, and tree size. One of the researchers [107] presented a fuzzy logic strategy based on greedy hill-climbing feature selection. The suggested classifier outperformed Naive Bayes, support vector machines, K-nearest neighbor, decision trees, and multilayer perceptron neural network classifiers in a comparative comparison.

In our research, we discovered that different languages face different obstacles based on the scope of the problem and the resources required. There have been several unique approaches to classification as well as earlier approaches with modifications and hybrid techniques, but still, there is a requirement for an efficient comprehensive classification model to deal with text in different languages. The next section of this paper talks about the results and discussion.

Results And Discussion

We have studied literature in twenty languages to find the answers to our research questions. In this section, we have presented a comparative analysis of five major languages. Table 1 shows some remarkable work done in different phases of Text Classification in these major languages. The next section of this paper talks about the conclusion and future scope of our study References.

Table: 1- *Comparative analysis of remarkable work done in different phases of Text Classification of five major languages*

Language(across) Phases of TC (below)	English	Arabic
Effect of pre-processing	Pre-processing has an impact on feature extraction, improves the performance of the classification model[15], and the right combination of pre-processing activities improves performance significantly[14]	Reducing the number of features lowered classifier complexity and space needs while also saving time[13], and emphasized the importance of pre-processing in emotion MLC [16]
Stop word removal	Smaller stop word list of 9 words performed similarly to a bigger list of 571 terms[34].	A finite State Machine-based algorithm was proposed[31]
Stemming	Some stemming algorithms performed better in one area while others performed better in another[42],ultrastemming approach significantly decreased the volume of representation[43]	The effectiveness of information retrieval is significantly increased when Arabic roots are used as indexing phrases. [45]
Lemmatization	linguistic pre-processing supported lemmatization[48],	Lemmatizer that gives a single lemma to each word in an Arabic sentence while taking into account the word context[48], a corpus of 2.2 million tokens was annotated and validated using machine learning and a lemmatization dictionary, and then it was used to train machine learning model [48]
Text Representation	The n-grams improved classifier performance[56], A bag-of-visual-words representation provide higher classification performance[57]adding sentences to unigrams improved the categorization outcome, training the Skip-gram and CBOW word vector models[60], employing multi-words for representation[61], statistical word co-occurrence network to represent text material[63], he best encoding strategy for convolutional neural networks was byte-level one-hot encoding.[65]	Representing Arabic Text semantically using Rich Semantic Graph (RSG) is one of the recent techniques that facilitate the process of manipulating the Arabic Language. [71].
Dimensionality Reduction	Unified manifold learning frameworks for semi-supervised and unsupervised dimension reduction were proposed[73], employed attribute overlap minimization and outlier elimination as dimensionality reduction strategy, and the CHI Square approach as a feature selection method[74]. The focus switched from feature engineering to learning with	Used the CHI Square approach as a feature selection method[74]. Fisher discriminant analysis was used[79],

Classification model used	probabilistic models like PLSA and LDA[81]. LSTM combined with convolutional layers produced the best results[85]. Hybrid ensemble classifiers using Random Forest performed significantly better than single classifiers[96]. One of the researchers [107] presented a fuzzy logic strategy based on greedy hill-climbing feature selection.	Proposed an artificial neural network with SVD for Arabic corpus [83], A novel hybrid Convolutional Genetic model was developed[]

Language(across) Phases of TC (below)	**Chinese**	**French**	**Spanish**
Effect of pre-processing	Using all the pre-processing steps increased the accuracy and performance of the CNN classification model [108]]	Because French's morphology is so richer than that of English, it requires special handling to obtain acceptable parsing performance. The parsing outcomes were slightly enhanced by training a lexicalized parser[109]	Lemmatization for short text datasets and stop word for long text datasets have a significant impact on the results of text classification[17]
Stop word removal	Chinese stop words contain very valuable information, so cannot be directly removed just like English stop words[15], Probabilistic automatic aggregated methodology-based algorithm [32]	Smaller stop word list of 9 words performed similarly to a bigger list of 571 terms[34].	Selective eradication of stop words in the term candidate was put out[33].
Stemming	The Ultrastemming approach significantly decreased the volume of representation [43]	The Ultrastemming approach significantly decreased the volume of representation[43]	300 stemming and reduction rules were paired with a dictionary search[37], ultrastemming approach significantly decreased the volume of representation[43]
Lemmatization	The building blocks in Chinese are called radicles since there is no concept of a stem. Because radicle separation would completely alter the meaning of the term, stemming and lemmatization are not	Use of TXM platform (http://textometrie.org) to lemmatize Medieval French literature[52]	The complexity of Spanish words is increased by prefixes. A neologism-aware lemmatizer can determine these prefixes[111].

	helpful for the Chinese language[110].		
Text Representation	The best encoding strategy for convolutional neural networks was byte-level one-hot encoding[65]	Adding sentences to unigrams improved the categorization outcome[58], an adaptive neural network technique based on the generalized Hebbian Algorithm to extract the first principal component [67]	q-gram tokenizers performed better than n-word tokenizers[59].
Dimensionality Reduction	Employed word embedding to calculate the most comparable terms to the current vocabulary determined by the IG algorithm and extend the lexicon with these words while adhering to specific rules[112].	The most comparable terms to the current vocabulary, as determined by the IG algorithm, were calculated using word embedding, and the lexicon was expanded using these words under specific constraints. In the end, this model produced positive experimental results in the Sogou and Fudan Chinese text categorization corpora[112].	PCA-based models get gains in RMSE of about 10%, whereas FSS-based models see benefits of about 25% in comparison to the benchmark. For the same endogenous variable, this benefit exceeds the 15% obtained by Vicente et al. (2015) [113].
Classification model used	Submitted their CNN-based model in July 2019 that simultaneously learns segmentation boundaries and stemming for the Myanmar language[93]. Authors [101] classified Chinese and Malay sentences and evaluated the effectiveness of their model against that of English.	Used recurrent convolutional neural networks[84]. Authors [90] used a combination model (LSTM+CNN) to classify French reviews and reported 93.7 percent accuracy.	To identify sarcasm in Spanish text, researchers [94] created a model based on the Transformer model's encoder component, Pre-trained versions of the RoBERTa-Large +CNN models were used[98]

Conclusions

Text categorization is an old but fascinating subject of study that has yielded promising findings but still needs to be investigated further to develop a generic model. In this paper, we examined the prior work of several scholars to discover answers to the following research questions.

RQ1: What are the technological challenges during the various stages of the categorization procedure that limit the building of a generic model?

Answer: We noticed that each phase of the text classification system incorporates a variety of strategies. There is a choice for picking among these tactics, but no explanations are provided, making it impossible to understand why a specific strategy should be chosen or rejected in a particular situation. Because of differing morphologies, different languages face unique obstacles, prompting academics to devise a unique solution for each language's pre-processing procedures. The lack of different datasets in a variety of languages also limits the ability to conduct experiments and validate research findings. There are several reasons for classifying the text under consideration, each of which necessitates a different approach and model.

RQ2. What is the best way to get a general model that can categorize, if not all, at least the most common types of unstructured text data in at least two or more languages?

Answer: Despite the variances indicated above, the overall analysis revealed a general pattern. For classifying text in other languages, scholars in other regions of the world use the same approaches and models. It gave us the impression that better classification results may be obtained by employing similar models, either by changing earlier models or hyper tuning them for better performance, or by devising some hybrid strategy, but there is still a need for a common model. According to our perspective, it may be possible to develop a generic model that can categorize text in a variety of languages by considering the following elements.

i) Morphology of many languages should be studied to comprehend the language's structure and grammar.
ii) Text documents should be translated into a non-language-specific intermediate format.
iii) To incorporate as much semantic information as possible, feature selection and text representation in a vector should be resilient.
iv) A hybrid classification model with layers representing diverse technologies should be preferred.

Future Scope

Our research found that there is a wide range of research potential connected to various stages of the Text Classification process in various languages. It is possible to analyse various pre-processing step combinations to determine a typical arrangement that works with texts in many languages. Rule-based tokenization may be a fruitful area of study. The Stop word elimination phase needs more research. Researchers might strive to create language independent solutions for stemming and lemmatization since they are dependent on a language morphology.

Future study will focus on feature extraction, feature selection, and document representation using these numerically valued features. A useful topic might be to incorporate more pertinent semantics in these representations. Researchers can come up with creative methods for creating new classifications that are effective and applicable to tiny datasets.

In addition to these areas, researchers could help develop and validate new datasets for various niche issues and make them accessible for use by other researchers. In order for people to comprehend the context of the suggested model or methodologies, the justifications for any methodology or method choice should be outlined in the literature.

Acknowledgments

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

Mahinovs, Aigars., Tiwari, Ashutosh., & Cranfield University, "Text classification method review", Cranfield University,2007.

Medagoda, N., Shanmuganathan, S., & Whalley, J., " A Comparative Analysis of Opinion Mining and Sentiment Classification in non-English Languages", 2013.

Jindal, R., Malhotra, R., & Jain, A., " Techniques for text classification: Literature review and current trends", vol. 12, Issue 2, 2015.

Faraz, A., "An Elaboration of Text Categorization and Automatic Text Classification Through Mathematical and Graphical Modelling", Computer Science & Engineering: An International Journal, Vol. 5(2/3), pp. 1–11,2015.

Jindal, R., Malhotra, R., & Jain, A., "Techniques for text classification: Literature review and current trends", vol. 12(2),2015.

Alabbas, W., Al-Khateeb, H. M., Mansour, A., "Arabic text classification methods: Systematic literature review of primary studies". Ieeexplore.Ieee.Org", pp.361–367,2016.

Thangaraj, M., of, M. S.-I. J., "Text classification techniques: a literature review". Search.Proquest.Com, Vol.13, pp.117–135,2018.

Kowsari, K., Heidarysafa, M., Brown, D. E., Meimandi, K. J., & Barnes, L. E., "RMDL: Random multimodel deep learning for classification. ACM International Conference Proceeding Series, pp.19–28,2018

Wahdan, A., Hantoobi, S., Salloum, S. A., &Shaalan, K., "A systematic review of text classification research based on deep learning models in Arabic language", vol.10(6), pp.6629–6643,2020.

Maw, M., Balakrishnan, V., Rana, O., &Ravana, S. D., "Trends and patterns of text classification techniques: A systematic mapping study. Malaysian Journal of Computer Science", vol.33(2), pp.102–117,2020.

Bhavani, A., & Santhosh Kumar, B., "A Review of State Art of Text Classification Algorithms".Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021, pp.1484–1490,2021.

Rakovac, I., Spat, S., Cadonna, B., Gütl, C., Leitner, H., Stark, G., & Beck, P., "Multi-label Text Classification of German Language Medical Documents", Academia.Edu.,2007.

Saad, M. K., "The impact of text preprocessing and term weighting on arabic text classification",2010.

Uysal, A., management, S. G.-I., "The impact of preprocessing on text classification", Elsevier,2022.

Kadhim, A. I., "Survey on supervised machine learning techniques for automatic text classification. Artificial Intelligence Review", vol.52(1), pp273-292,2019.

Mulki, H., Ali, C. B., Haddad, H., &Babaoˇ Glu, I., "Preprocessing Impact on Multi-label Emotion Classification", Tw-StAR at SemEval-2018, pp.167–171,2022.

Orellana, G., Arias, B., Orellana, M., Saquicela, V., Baculima, F., & Piedra, N., "A study on the impact of pre-processing techniques in Spanish and english text classification over short and large text documents". In 2018 International Conference on Information Systems and Computer Science (INCISCOS), pp. 277-283,2018.

Kozhevnikov, V., &Pankratova, E., "Research Of Text Pre-Processing Methods For Preparing Data In Russian For Machine Learning".2020.

Saitov, K., &Derczynski, L., "Abusive Language Recognition in Russian", In Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing, pp. 20-25,2021.

Graves A., "Generating Sequences With Recurrent Neural Networks", arXiv. 2013 Aug 4;abs / 1308.0850.

Webster JJ, Kit C., "Tokenization as the Initial Phase in NLP". In: Proceedings of the 14th C onference onComputational Linguistics, COLING'92, Nantes, France: Association for Computational Linguistics, vol.4, pp. 110A1110,1992.

Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, et al., "Moses: OpenSourceToolkit for Statistical Machine Translation", In Proceedings of the 45th Annual Meeting of the Association for Computational Lingui stics Companion Volume Proceedings of the Demo and Poster Sessions;2007 Jun. Pra gue, Czech Republic: Association for Computational Linguistics. pp. 177—180,2007.

spacy.io," Linguistic featuresTokenization", 2022. [cited 2022 Apr 13].

Mielke SJ, Alyafeai Z, Salesky E, Raffel C, Dey M, Gallé M, et al.,
" Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP",2021.

Tokenizer summary,Available at: https:// huggingface.co/transformers/v3.0.2/tokenizer sum mary.html., accessed August 2022.

Sennrich R, Haddow B, Birch A.," Neural Machine Translation of Rare Words with Subword Units", In:
Proceedings of the 54th Annual Meeting of the Association for Computational Lingui stics Berlin, Germany: Association for Computational Linguistics, vol.1 pp. 171a 1725,2016.

Gage P.," A New Algorithm for Data Compression", The C Users J. 1994 Feb; vol.12(2):2W 38.

Wang C, Cho K, Gu J.," Neural Machine Translation with Byte-vel Subwords", vol. 34, pp. 7—12,2020 Feb.

Schuster M, Nakajima K.," Japanese and Korean voice search", In: 2012 IEEE International Conference
on Acoustics, Speech and Signal Processing (ICASSP); 2012 Mar 2W30. Kyoto, Japa n. pp. 5149— 5152,2012.

Kudo T. Subword Regularization: Improving Neural Network Translation Models with Multi ple SubwordCandidates. In: Proceedings of the 56th Annual Meeting of the Associati on for Computational Linguistics (Volume 1: Long Papers); 2018 Jul 1W20. Melbourne, Australia: Association for Computational Linguistics. pp. 6A75.

Al-Shalabi, R., Kanaan, G., Jaam, J. M., Hasnah, A., &Hilat, E., " Stop-word removal algorithm for Arabic language", pp.545–545,2004.

Zou, F., Wang, F. L., Deng, X., Han, S., & Wang, L. S., "Automatic construction of Chinese stop word list", In Proceedings of the 5th WSEAS international conference on Applied computer science, Stevens Point, WI, USA: World Scientific and Engineering Academy and Society (WSEAS), pp. 1010-1015,2006.

Barrón-Cedeno, A., Sierra, G., Drouin, P., &Ananiadou, S., "An improved automatic term recognition method for Spanish", In International Conference on Intelligent Text Processing and Computational Linguistics, pp. 125-136,2009.

Dolamic, L., & Savoy, J., "When stopword lists make the difference". Journal of the American Society for Information Science and Technology, vol.61(1), pp.200–203,2010.

Saini, J. R., Raulji, J. K., Director, C., Supervisor, R., & Ambedkar, B., "Stop-word removal algorithm and its implementation for Sanskrit language", Researchgate.Net, vol.150(2), pp.975–8887,2016.

Buttar, P., Kaur, J., & Buttar, P. K., "A Systematic Review on Stopword Removal Algorithms",2018.

Honrado, A., Leon, R., O'Donnel, R., & Sinclair, D., "A word stemming algorithm for the Spanish language", In Proceedings Seventh International Symposium on String Processing and Information Retrieval. SPIRE 2000, pp. 139-145,2000. IEEE.

Basili, R., Moschitti, A., &Pazienza, M. T., "A Hybrid Approach to Optimize Feature Selection Process in Text Classification", Springer-Verlag, 2001.

Vinokourov, A., Shawe-Taylor, J., &Cristianini, N., " Inferring a Semantic Representation of Text via Cross-Language Correlation Analysis".

Asian, J., Williams, H., Twenty, S. T.-P., "Stemming indonesian", Citeseer. Retrieved April 16, 2022,

Majgaonker, M. M., "Discovering suffixes: A Case Study for Marathi Language", IJCSE) International Journal on Computer Science and Engineering, vol.02(08), pp.2716–2720,2010.

Anjali, M., & Jivani, G. (n.d.). A Comparative Study of Stemming Algorithms. www.ijcta.com

Torres-Moreno, J.-M., "Beyond Stemming and Lemmatization: Ultra-stemming to Improve Automatic Text Summarization", 2012.

Hafhizah Abd Rahim, N., Abdullah, Z., Ismailov, A., Abdul Jalil, M., & Abd Rahim, N., "The Development of the Uzbek Stemming Algorithm", Article in Journal of Computational and Theoretical Nanoscience, 2017.

Zeroual, I., &Lakhouaja, A., " Arabic information retrieval: Stemming or lemmatization", In 2017 Intelligent Systems and Computer Vision (ISCV), pp. 1-6, 2017. IEEE.

Rahutomo, R., Lubis, F., Muljo, H. H., &Pardamean, B.," Preprocessing Methods and Tools in Modelling Japanese for Text Classification", Proceedings of 2019 International Conference on Information Management and Technology, ICIMTech 2019, pp.472–476, 2019.

Moral, C., de Antonio, A., Imbert, R., & Ramírez, J., "A survey of stemming algorithms in information retrieval", Information Research: An International Electronic Journal, vol.19(1), 2014.

Basili, R., Moschitti, A., &Pazienza, M. T., "A Hybrid Approach to Optimize Feature Selection Process in Text Classification", Springer-Verlag,2001.

Chrupała, G., "Simple data-driven context-sensitive lemmatization,2006.

Paul, S., Tandon, M., Joshi, N., & Mathur, I., "Design of a Rule Based Hindi Lemmatizer", pp. 67–74,2013.

Joulin, A., Grave, E., Bojanowski, P., &Mikolov, T., " Bag of Tricks for Efficient Text Classification",2016.

Lavrentiev, A., Heiden, S., &Decorde, M., "Building an open morphological lexicon and lemmatizing old French texts with the TXM platform. In Corpus linguistics-2017, pp. 48-52,2017.

Freihat, A. A., Abbas, M., Bella, G., &Giunchiglia, F., "Towards an optimal solution to lemmatization in arabic", Procedia computer science, vol.142, pp132-140,2018.

Boudchiche, M., &Mazroui, A., "A hybrid approach for Arabic lemmatization. International Journal of Speech Technology, vol.22(3), pp.563-573, 2019.

Leopold, E., "Text Categorization with Support Vector Machines. How to Represent Texts in Input Space? ", vol. 46, 2002.

Šilić, A., Chauchat, J. H., Bašić, B. D., & Morin, A. (2007). N-grams and morphological normalization in text classification: A comparison on a Croatian-English parallel corpus. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4874 LNAI, 671–682.

Yang, J., Jiang, Y. G., Hauptmann, A. G., & Ngo, C. W., " Evaluating bag-of-visual-words representations in scene classification. Proceedings of the ACM International Multimedia Conference and Exhibition,2007.

D'hondt, E., Verberne, S., Koster, C., &Boves, L., "Text representations for patent classification. Computational Linguistics", vol.39(3), pp.755-775,2013.

Tellez, E. S., Miranda-Jiménez, S., Graff, M., Moctezuma, D., Siordia, O. S., &Villaseñor, E. A., "A case study of Spanish text transformations for twitter sentiment analysis", Expert Systems with Applications, vol.81, pp. 457-471,2017.

Joulin, A., Grave, E., Bojanowski, P., &Mikolov, T., " Bag of Tricks for Efficient Text Classification", 2016.

Wang, G., Li, C., Wang, W., Zhang, Y., Shen, D., Zhang, X., Henao, R., & Carin, L., "Joint Embedding of Words and Labels for Text Classification", 2018.

Mouriño-García, M. A., Pérez-Rodríguez, R., Anido-Rifón, L., &Vilares-Ferro, M., "Wikipedia-based hybrid document representation for textual news classification. Soft Computing", vol.22(18), pp.6047–6065,2018.

Nikolentzos, G., &Vazirgiannis, M., "Learning structural node representations using graph kernels", IEEE transactions on knowledge and data engineering, vol.33(5), pp.2045-2056,2019.

Kumar, S., Wintner, S., Smith, N. A., &Tsvetkov, Y., "Topics to avoid: Demoting latent confounds in text classification", EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference.2020.

Zhang, J., Li, Y., Tian, J., Advanced, T. L.-2018 I. 3rd, & 2018, undefined. (n.d.). LSTM-CNN hybrid model for text classification. Ieeexplore.Ieee.Org. Retrieved May 9, 2022, from

Virtanen, A., Kanerva, J., Ilo, R., Luoma, J., Luotolahti, J., Salakoski, T., Ginter, F., &Pyysalo, S., "Multilingual is not enough: BERT for Finnish", Retrieved February 28, 2022.

Delichere, M., & Memmi, D., "Neural dimensionality reduction for document processing", In ESANN, pp. 211-216,2002.

Liu, N., Zhang, B., Yan, J., Chen, Z., Liu, W., Bai, F., &Chien, L., "Text representation: From vector to tensor. In Fifth IEEE International Conference on Data Mining (ICDM'05), pp.4,2005. IEEE.

Mehler, A., Geibel, P., &Pustylnikov, O., "Structural Classifiers of Text Types: Towards a Novel Model of Text Representation", In LDV Forum (Vol. 22, No. 2, pp. 51-66, 2007.

Khan, A., Baharudin, B., & Khan, K., "Semantic based features selection and weighting method for text classification", In 2010 international symposium on information technology, vol. 2, pp. 850-855, 2010. IEEE.

Ismail, S., Moawd, I., &Aref, M., " Arabic text representation using rich semantic graph: A case study. In Proceedings of the 4th European conference of computer science (ECCS'13), pp. 148-153,2013.

Mesleh, A., Moh'd, A., &Mesleh, A. (2007). Chi Square Feature Extraction Based Svms Arabic Language Text Categorization System. Journal of Computer Science, 3(6), 430–435.

Nie, F., Xu, D., Tsang, I. W. H., & Zhang, C., "Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction", IEEE Transactions on Image Processing, vol.19(7), pp.1921-1932, 2010.

Fong, S., &Cerone, A., "Attribute overlap minimization and outlier elimination as dimensionality reduction techniques for text classification algorithms", Journal of Emerging Technologies in Web Intelligence, vol.4(3), vol. 259-263,2012.

Gasanova, T., Sergienko, R., Semenkin, E., &Minker, W., " Dimension reduction with coevolutionary genetic algorithm for text classification", In 2014 11th International Conference on Informatics in Control, Automation and Robotics (ICINCO), vol.1, pp. 215-222,2014.

Zhang, Y., Lease, M., on, B. W.-P. of the A. C., "Active discriminative text representation learning". Ojs.Aaai.Org. Retrieved April 19, 2022,

Zhu, L., Wang, G., & Zou, X., "Improved information gain feature selection method for Chinese text classification based on word embedding", ACM International Conference Proceeding Series, pp.72–76,2017.

Walkowiak, T., Datko, S., &Maciejewski, H., "Feature Extraction in Subject Classification of Text Documents in Polish", Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10842 LNAI, 445–452,2018.

AbuZeina, D., & Al-Anzi, F. S., "Employing fisher discriminant analysis for Arabic text classification", Computers & Electrical Engineering, vol.66, pp.474-486,2018.

Puri, S., " A Review on Dimensionality Reduction in Fuzzy-and SVM-Based Text Classification Strategies", In Congress on Intelligent Systems, pp. 613-631,2020.

Ramalingam, V. V., Pandian, A., Chetry, P., & Nigam, H., "Automated essay grading using machine learning algorithm", In Journal of Physics: Conference Series, vol. 1000(1), pp. 012030,2018.

Walkowiak, T., & Malak, P., "Polish Texts Topic Classification Evaluation", 2018.

Harrag, F., & El-Qawasmah, E., "Neural Network for Arabic text classification", In 2009 Second International Conference on the Applications of Digital Information and Web Technologies, pp.778-783, 2009. IEEE.

Lai, S., Xu, L., Liu, K., & Zhao, J., "Recurrent convolutional neural networks for text classification", In Twenty-ninth AAAI conference on artificial intelligence, 2015.

Johnson, R., & Zhang, T., "Supervised and Semi-Supervised Text Categorization using LSTM for Region Embeddings",2016.

Johnson, R., & Zhang, T., "Deep pyramid convolutional neural networks for text categorization. ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), vol.1, pp. 562–570,2017.

Wang, G., Li, C., Wang, W., Zhang, Y., Shen, D., Zhang, X., Henao, R., & Carin, L., "Joint Embedding of Words and Labels for Text Classification",2018.

Stab, C., Miller, T., Schiller, B., Rai, P., &Gurevych, I., "Cross-topic Argument Mining from Heterogeneous Sources", pp. 3664–3674.

Adhikari, A., Ram, A., Tang, R., & Lin, J., "DocBERT: BERT for Document Classification", 2019.

Habbat, N., Anoun, H., &Hassouni, L., "LSTM-CNN Deep Learning Model for French Online Product Reviews Classification", In International Conference on Advanced Technologies for Humanity, pp. 228-240,2021. Springer, Cham.

Jiang, S., Pang, G., Wu, M., &Kuang, L. (2012). An improved K-nearest-neighbor algorithm for text categorization. Expert Systems with Applications, vol.39(1), pp.1503-1509,2012.

Zhang, J., Li, Y., Tian, J., Advanced, T. L., "LSTM-CNN hybrid model for text classification", 2018.

Oo, Y., &Soe, K. M., " Better pretrained embedding with convolutional neural networks for morphological stemming", In Proceedings of the 2019 3rd International Conference on Artificial Intelligence and Virtual Reality, pp. 60-64,2019.

González, J. Á., Hurtado, L. F., &Pla, F.,"ELiRF-UPV at IroSvA: Transformer Encoders for Spanish Irony Detection", In IberLEF@ SEPLN, pp. 278-284,2019.

Liu, Y., Ma, J., Tao, Y., Shi, L.,L. W, " Hybrid Neural Network Text Classification Combining TCN and GRU", Ieeexplore.Ieee.Org. Retrieved May 9, 2022,

Sainin, M., Alfred, R., and, F. A.-J. of I., "Ensemble Meta Classifier With Sampling And Feature Selection For Data With Imbalance Multiclass Problem", E-Journal.Uum.Edu.My, vol.20(2), pp.103–133,2021.

Alsaleh, D., &Larabi-Marie-Sainte, S., "Arabic text classification using convolutional neural network and genetic algorithms. IEEE Access, 9, pp.91670-91685,2021.

Guan, Z., "TSIA team at FakeDeS 2021: Fake News Detection in Spanish Using Multi-Model Ensemble Learning", In IberLEF@ SEPLN, pp. 661-667,2021.

López-García, G., Jerez, J. M., Ribelles, N., Alba, E., &Veredas, F. J.,"Transformers for clinical coding in spanish. IEEE Access, 9, pp.72387-72397,2021.

Bloehdorn, S., Cimiano, P., Hotho, A., Forum, S. S.-L., "An Ontology-based Framework for Text Mining",2005.

Zhang, Y., Tsai, F. S., &Kwee, A. T.,"Multilingual sentence categorization and novelty mining. Information processing & management, vol.47(5), pp.667-675,2011.

Kowsari, K., Heidarysafa, M., Brown, D. E., Meimandi, K. J., & Barnes, L. E., "RMDL: Random multimodel deep learning for classification", ACM International Conference Proceeding Series, pp.19–28,2018.

El-Alami, F. Z., El Mahdaouy, A., El Alaoui, S. O., & En-Nahnahi, N., "A deep autoencoder-based representation for Arabic text categorization", Journal of Information and Communication Technology, vol.19(3), pp.381-398,2020.

Kim, J., Jang, S., Park, E., & Choi, S., "Text classification using Capsules", Neurocomputing, vol.376,pp. 214-221, 2020.

Jeon, H. K., & Cheong, Y. G., "A peer learning method for building robust text classification models", Proceedings - 2021 IEEE International Conference on Big Data and Smart Computing, BigComp 2021, pp.321–324, 2021.

Kumar, S., Ratnoo, S., &Vashishtha, J., " Hyper Heuristic Evolutionary Approach For Constructing Decision Tree Classifiers", Journal of Information and Communication Technology, vol.20(2), pp.249-276, 2021.

Al-Behadili, H. N. K., & Ku-Mahamud, K. R., "Fuzzy unordered rule using greedy hill climbing feature selection method: An application to diabetes classification", Journal of Information and Communication Technology, vol.20(3),2021.

Li, A., & Chen, Y., "re-processing Analysis for Chinese Text Sentiment Analysis", In Proceedings of the 2017 2nd International Conference on Communication and Information Systems, pp. 318-323,2017.

Seddah, D., Chrupała, G., Çetinoğlu, Ö., Van Genabith, J., &Candito, M.,"Lemmatization and lexicalized statistical parsing of morphologically-rich languages: the case of french", In Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages, pp.85-93,2010.

Chinese Natural Language (Pre)processing: An Introduction | by Sidney Kung | Towards Data Science

Zenón Hernández-Figueroa, Francisco J. Carreras-Riudavets, Gustavo Rodríguez-Rodríguez, "Automatic syllabification for Spanish using lemmatization and derivation to solve the prefix's prominence issue, Expert Systems with Applications", vol.40(17), pp. 7122-7131, 2013,

Zhu, L., Wang, G., & Zou, X., "Improved information gain feature selection method for Chinese text classification based on word embedding", In proceedings of the 6th International Conference on Software and Computer Applications ,pp. 72-76,2017.

Mulero, R., & García-Hiernaux, A.,"Forecasting Spanish unemployment with Google Trends and dimension reduction techniques", SERIEs, vol.12(3), pp.329-349,2021.