Wiley
International Journal of Intelligent Systems
Volume 2025, Article ID 6091900, 40 pages
https://doi.org/10.1155/int/6091900

WILEY

Review Article

Artificial Intelligence for Text Analysis in the Arabic and Related Middle Eastern Languages: Progress, Trends, and Future Recommendations

Abdullah Y. Muaad,[1] **Md Belal Bin Heyat**,[2] **Faijan Akhtar**,[3] **Usman Naseem**,[4] **Wadeea R. Naji**,[5] **Suresha Mallappa**,[6] **and Hanumanthappa J.**[6]

[1]*IT Department, Sana'a Community College, Sana'a, Yemen*
[2]*CenBRAIN Neurotech Center of Excellence, School of Engineering, Westlake University, Hangzhou, China*
[3]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, China*
[4]*School of Computer Science, University of Sydney, Sydney, Australia*
[5]*Department of Computer Science & Information Technology, Ibb University, Ibb, Yemen*
[6]*Department of Studies in Computer Science, University of Mysore, Mysore, Karnataka, India*

Correspondence should be addressed to Abdullah Y. Muaad; abdullahmuaad9@gmail.com, Md Belal Bin Heyat; belalheyat@westlake.edu.cn, and Faijan Akhtar; faijanakhtar98@gmail.com

Received 3 October 2024; Accepted 13 May 2025

Academic Editor: Mohamadreza (Mohammad) Khosravi

In the last 10 years, there has been a rise in the number of Arabic texts, which necessitates a more profound understanding of algorithms to efficiently understand and classify Arabic texts in many applications, like sentiment analysis. This paper presents a comprehensive review of recent developments in Arabic text classification (ATC) and Arabic text representation (ATR). We analyze the effectiveness of various models and techniques. Our review finds that while deep learning models, particularly transformer-based architectures, are increasingly effective for ATC, challenges such as dialectal variations and insufficient labeled datasets remain key obstacles. However, developing suitable representation models and designing classification algorithms is still challenging for researchers, especially in Arabic. A basic introduction to ATC is provided in this survey, including preprocessing, representation, dimensionality reduction (DR), and classification with many evaluation metrics. In addition, the survey includes a qualitative and quantitative study of the ATC's existing works. Finally, we conclude this work by exploring the limitations of the existing methods. We also mention the open challenges related to ATC, which help researchers identify new directions and challenges for ATC.

Keywords: AI for language; AI for text; Arabic; artificial intelligence; dimensionality reduction; natural language processing; text classification; text representation

1. Introduction

Nearly 447 million people speak Arabic as their first language. At the same time, it is one of the world's most widely spoken languages and is regarded as the fourth official language of the United Nations (UN) [1, 2]. The rise in the number of users leads to an increase in Arabic textual data generated daily. So, extracting information from such huge

data is a challenging task, especially for Arabic text (AT). Therefore, there is a need to preprocess AT and remove words that do not have significant meaning, change words to their roots, eliminate noise, and improve the performance of Arabic text classification (ATC) [3].

The process of cleaning and preparing text for further processing is known as preprocessing. It is the preliminary step in any text classification pipeline. Specific preprocessing methods and algorithms are required to extract useful patterns from unstructured Arabic textual data. Preprocessing for AT includes many techniques such as white space removal, lemmatization, stemming, and stop-word removal. There are several preprocessing techniques that have been used to enhance the performance of ATC. However, most of the available techniques are still not able to cover all the requirements to prepare AT for further processing due to the complexity of AT [2].

Representation and feature engineering (selection and extraction) are the second steps in the ATC pipeline. The efficiency of succeeding natural language processing (NLP) tasks is strongly influenced by the quality of these techniques [4]. Representation is the process of converting unstructured text documents into their structured equivalent so that machine learning (ML) algorithms [5, 6] can understand [7]. Several feature extraction techniques, including bag-of-words (BoW) [8], term frequency–inverse document frequency (TF–IDF) [9], term class relevance (TCR) [10], term class weight–inverse class frequency (TCW–ICF) [10, 11], symbolic representation, and N-gram features, have been used for feature representation. At the same time, different levels of representation can be used to represent text with different levels, such as character-level, word-level, and phrase-level representations [8, 9].

Most of the researchers have used TF–IDF or BoW, which are inherently problematic due to the lack of the sequence of the words and skip the semantics meaning of the sentence, so various sentences might have the same vector if they have the same words with a different sequence, for example, علي مدرس (which means Ali is a teacher) and أعلي مدرس؟ (which means Dose Ali a teacher?). However, these techniques do not have problems with memory consumption for storage, but they lose semantic meaning. To overcome those limitations, many other techniques have been proposed, for example, Word2Vec [10], GloVe (https://nlp.stanford.edu/projects/glove/) [11], and contextualized word representations [12].

Text categorization is the process of determining if a text belongs to one of the several predefined categories based on its meaning [13, 14]. Once the representation of a given text is achieved, a classifier needs to classify AT into various classes [15]. Many of the ML algorithms such as Decision Trees (DT) [16, 17], Naive Bayes (NB) [18, 19], support vector machines (SVM) [20, 21], and artificial neural networks (ANN) [21, 22] were used for ATC. However, getting high performance is still a real challenge. Therefore, in this survey, we attempt to perform a comprehensive taxonomy study for ATC to find the strengths and weaknesses of the existing work.

Given the growing demand for accurate ATC in domains such as healthcare, finance, and e-commerce, it is essential to explore effective techniques, address linguistic challenges, and mitigate ethical concerns. This study aims to provide a comprehensive taxonomy survey of ATC, analyze existing approaches, and highlight open research challenges and future directions to improve the field. Due to the limited research on ethical considerations and bias in ATC, existing studies have not sufficiently addressed this aspect, making it a key future challenge for researchers. Therefore, there is a pressing need to advance research in fairness, transparency, and explainability in Arabic NLP systems to ensure the development of more equitable and accurate models that meet the requirements of various real-world applications.

1.1. Motivation. Due to the increase in the number of ATs in social media, there is a need to perform a comprehensive study and analysis to find the strengths and weaknesses of existing studies on ATC, which helps build an efficient, effective, and robust algorithm to represent and classify AT. Simultaneously, this increased the number of Arabic speakers to more than 447 users and increased the number of internet users by 9348% more than English (https://www.internetworldstats.com/stats7.htm), which has only 7429%. Therefore, developing tools and applications to handle AT became mandatory. The following is a list of some motivations for this survey:

- Increase the number of Arabic users and text generation for the Arabic language in many domains, especially with COVID-19.
- Many researchers still use traditional representation techniques such as a BoW that cannot work well with huge.
- Little research is conducted on AT compared to other languages, such as English.
- Lack of tools and applications for the Arabic language.
- Many non-Arab people who speak and use the Arabic language as a second language are also more than native speakers; therefore, studying these limitations and finding solutions for these problems and challenges will help many people.

1.2. Contributions. The main contributions of this survey are mentioned in the following list:

- A comprehensive review of available studies and existing surveys in ATC, focusing on their objective, scopes, and research gaps.
- Explores the architecture of ATC and ATR.
- A comparative study of ATC stages such as preprocessing, representation, feature engineering, and classification.
- A comparative study of seven ATC and ATR models to evaluate their performance through an experimental analysis using the AlKhaleej dataset.

- A quantitative analysis of the proposed techniques for ATC based on publication year and categories.
- Review and mention the available datasets and open-source libraries.
- Implementation and discussion for seven models based on preprocessing, feature selection, feature extraction, and classification algorithms such as NB and SVC to evaluate their performance.
- A qualitative analysis of the ATC and ATR models based on their strengths and weaknesses.
- An overview of current challenges and future research work after quantitative analysis.

While there have been several surveys on ATC and ATR, most of them focus on limited aspects such as preprocessing techniques or specific classification algorithms. This work offers a broader perspective by providing a comprehensive taxonomy that encompasses all stages of ATC, including preprocessing, representation, dimensionality reduction (DR), and evaluation. Furthermore, this work uniquely combines qualitative and quantitative analyses, which offer deeper insights into the strengths and limitations of existing methods. Unlike previous works, this survey also emphasizes the challenges specific to Arabic language features, such as its complex morphology and dialectical variations, and provides actionable recommendations for overcoming these challenges. Such a holistic approach has not been addressed in existing literature, making this study a novel and valuable contribution to the field. In addition, this article claims to increase the efficiency of learning cutting-edge methodologies for ATC. In addition, it identifies prospective research gaps, allowing researchers to pick their research routes. According to our information, it will enlarge their minds and open the path for future new approaches.

1.3. Organization of the Paper. The organization of this survey is as follows: Section 2 studies and compares the existing surveys. Section 3 discusses the background and general architecture of the ATC model. The main steps of ATC are explored and analyzed in Sections 4, 5, 6, 7, and 8. The tools and open-source library are presented in Section 9. The quantitative analysis is highlighted in Section 10. The experimental analysis is presented in Section 11. The discussion and open challenges are highlighted in Sections 12 and 13. Finally, we conclude this survey in the conclusion section. For more clarity, Figure 1 illustrates this taxonomy using a mind map diagram using the lucid chart.

2. Existing Surveys

One of the pivotal goals of the article is to explore the existing surveys. However, some surveys have been done for ATC. This survey is examined, assessed, and compared with existing surveys in this section. We are inspired to survey all steps to make this research different from the existing one. There has been a slew of reviews and polling pieces published for ATC. However, most of them do not study each step individually. At the same time, in comparison with the previous work, this section will study the prior surveys on ATC and will provide an analysis comparing other researchers' work with this taxonomy. In the next part, we study them and compare them. As shown in Table 1, there are various extant reviews and surveys in the state of ATC. However, they did not consider all stage aspects, as our study did.

A critical review of the methodologies revealed that while traditional techniques such as BoW and TF–IDF excel in simplicity and efficiency, they struggle with sparsity and fail to capture semantic relationships in AT. Similarly, deep learning (DL) methods, particularly transformer-based models like BERT, show promising results but require substantial training data, often unavailable for dialectical Arabic. The reviewed studies highlight a recurring limitation: the inability of existing models to adapt to Arabic's morphological complexity and dialectical diversity. Addressing these challenges necessitates the development of more context-aware models and larger annotated datasets.

3. General Architecture of ATC

This part describes the entire ATC workflow, as shown in Figure 2, as well as a simple notion of preprocessing, representation, and classification models in Sections 3.1, 3.2, 3.3, and 3.4, respectively.

3.1. Preprocessing. The process of cleaning and preparing the text for subsequent processing is known as preprocessing. It is the initial step in the text categorization pipeline [35]. Tokenization, stop-word removal, and stemming are only a few of the methods for text preparation. Tokenization is a method of removing white space and special characters from a document. Stop words are general terms employed to complement informational material with minimal meaning; they provide a grammatical function but do not reveal the subject matter, and there are many other techniques [36].

3.2. Representation. Text representation is a crucial stage in any text classification model. ML algorithms [37, 38] can understand the transformation of unstructured text into structured text documents. There are different types of representation (level) of text, such as character level, word level, sentence level, phrase level, document level, and so on. The most important thing here is not only representation; feature engineering (selection and extraction) is also significant in making the ATC system work efficiently and effectively.

3.3. DR and Feature Engineering. DR is employed to reduce the dimensionality of the input feature space. There are various methods to reduce the size, such as feature selection (wrapper, embedded, filter), ensemble, and hybrid techniques. DR can be applied simultaneously in the preprocessing phase, such as stemming before or after representation, such as chi-square.

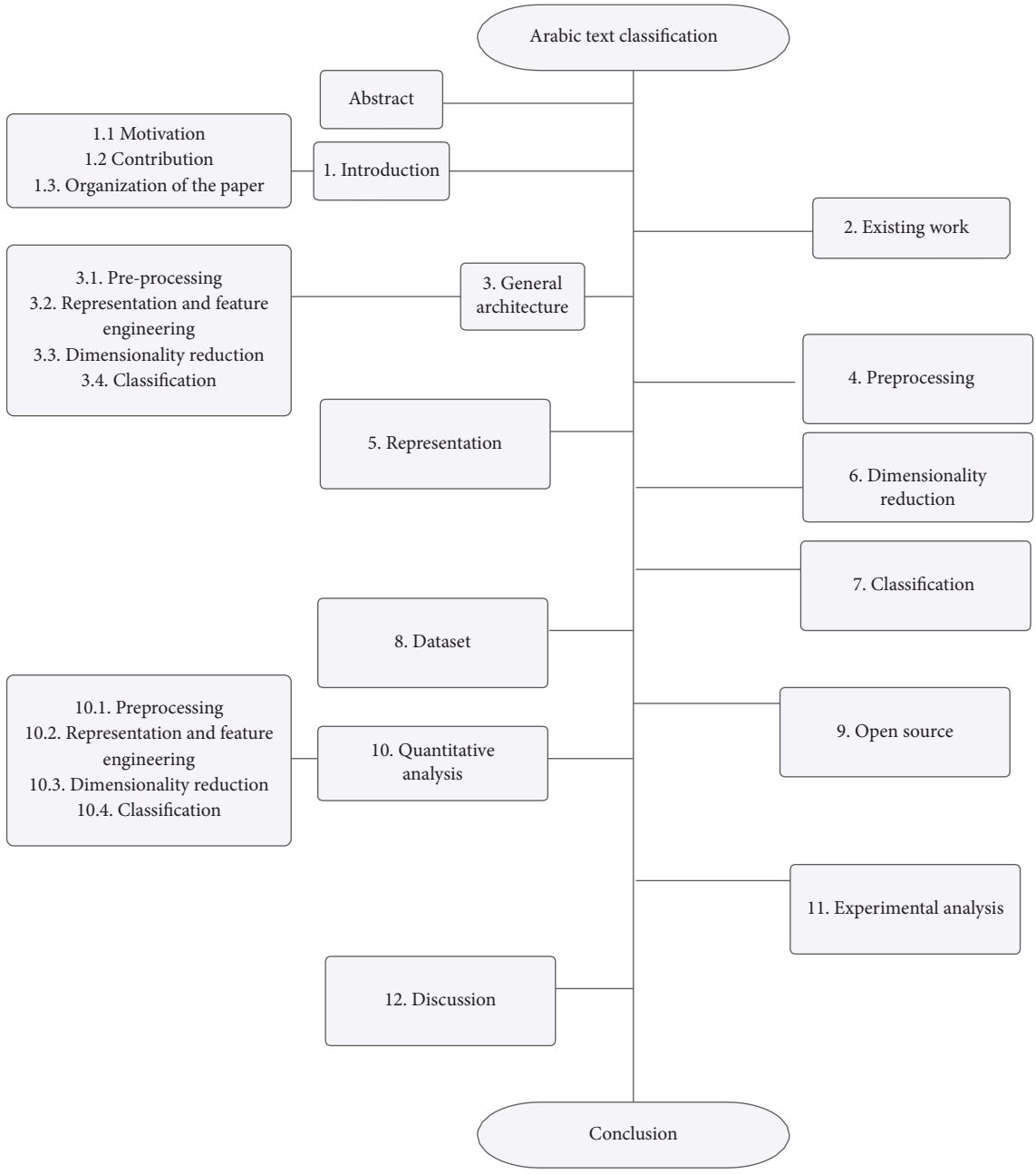

Figure 1: Mind map diagram for the taxonomy of Arabic text classification.

3.4. Classification. Once the representation for a given text collection is created through an optimal set of representation and feature extraction techniques, the classifier has to be trained to learn the pattern of classifying text into different classes [15]. There are many applications for text classification [39, 40] in other scenarios such as information retrieval (IR), sentiment analysis (SA), recommender systems, and hate speech detection. At the same time, text classification can be utilized in numerous domains such as health, social sciences, and law domains [41, 42].

4. Preprocessing

Preprocessing techniques prepare text for further processing by transforming unstructured text into structured data. Many techniques have been used for this task. Figure 3 and Table 2 explore these techniques based on work that has been done for ATC. Each preprocessing method in ATC has its advantages and disadvantages, impacting model performance in different ways. For instance, diacritic removal simplifies text representation and reduces data sparsity, but it may lead to ambiguity, as

Table 1: Comparative analysis of our survey with the existing survey in ATC.

Ref.	Year	Preprocessing	Features extraction	Classification	Qualitative analysis	Taxonomy	Experimental analysis	Quantitative analysis	Evaluation matrices
[23]	2016	✓	✓	✓					
[24]	2017	✓	✓	✓					
[25]	2017		✓	✓					✓
[26]	2018			✓					
[27]	2019	✓		✓					✓
[28]	2019		✓	✓	✓	✓			✓
[29]	2019	✓	✓	✓	✓				✓
[30]	2019		✓	✓		✓	✓		✓
[31]	2020	✓	✓	✓	✓				✓
[32]	2020	✓	✓	✓					✓
[2]	2021		✓	✓	✓		✓		✓
[33]	2022	✓			✓	✓	✓	✓	✓
[34]	2023	✓	✓					✓	✓
This work		✓	✓	✓	✓	✓	✓	✓	✓

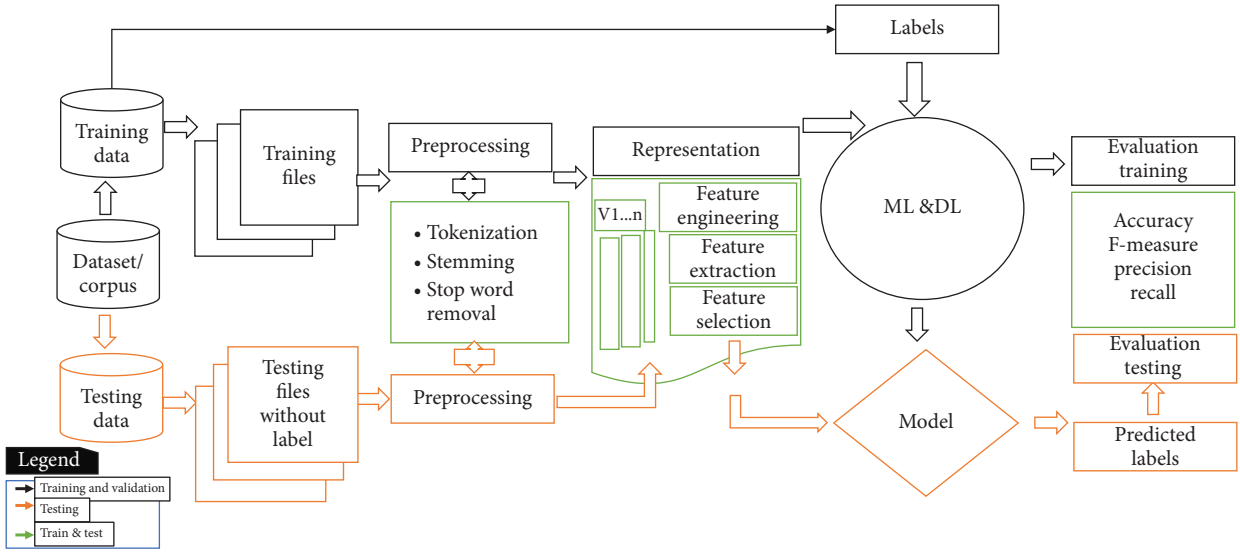

Figure 2: Arabic text classification architecture.

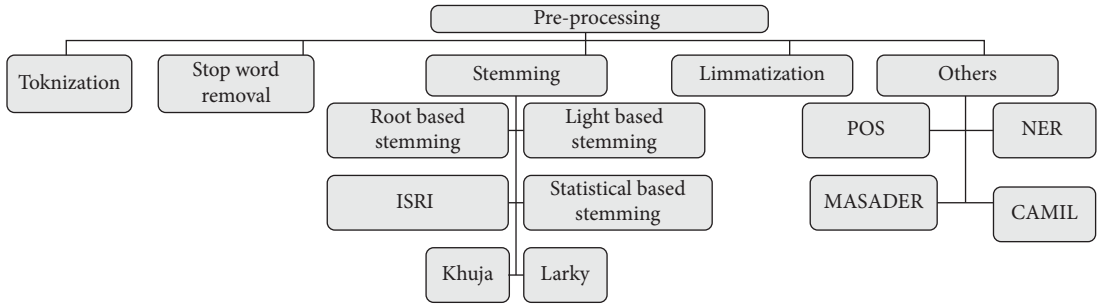

Figure 3: Preprocessing techniques for Arabic text classification.

some words have different meanings depending on diacritics. Stemming and lemmatization help normalize words by reducing them to their root forms, improving generalization; however, stemming can be overly aggressive, cutting words too short, and losing meaning, while lemmatization requires linguistic knowledge and is computationally expensive. Tokenization, especially in Arabic, is challenging due to the absence of clear word boundaries in certain cases, which may lead to errors in splitting words. Stop-word removal helps reduce computational complexity and improve efficiency, but in some contexts, stop words carry semantic importance, and their removal can affect classification accuracy. Normalization techniques, such as unifying different forms of Arabic letters (e.g., converting "ي" to "ى"), improve consistency but may lead to unintended modifications in certain words. Therefore, selecting the right preprocessing techniques requires balancing efficiency, linguistic integrity, and task-specific requirements to optimize the performance of ATC.

4.1. Tokenization. It is the process of segmenting a given text into small units. Alyafeai et al. proposed three novel text tokenization algorithms for AT [36].

4.2. Linguistic Preprocessing. It refers to additional preprocessing such as part-of-speech tagging, which is applied to get additional information about the content of the text, for instance, ADIDA, MADAMIRA, etc. [89].

4.3. Stop-Word Removal. It refers to the elimination of words that do not give meaning to the text. Auxiliary words, prepositions, conjunctions, modal words, and other high-frequency words in diverse publications are all examples of stop words [82].

Table 2: Comparative analysis of preprocessing techniques.

Ref.	Year	Objective	Method	Dataset	Evaluation matrices
[43]	2001	Aim to improve normalization and stemming	Normalization and stemming	—	Precision
[44]	2002	To design new light stemmers	Stemming	—	Precision
[45]	2002	To improve retrieval effectiveness	Light stemming approach	Provided by the text retrieval conference	Precisions and recall
[46]	2007	The goal is to contrast and compare two feature selection techniques. Light stemming vs. stemming	Stem vectors and light stem vectors	15,000 documents for three classes	F1-score
[47]	2008	To generate index words for AT documents	Stemming and weight assignment technique, and an autoindexing method	24 arbitrary texts of different lengths	Recall and precision
[48]	2008	To introduce a novel lemmatization algorithm	Lemmatization	House corpus	Recall and precision
[49]	2008	Proposed a new method for stemming AT	Stemming techniques	—	—
[50]	2008	Design a new stemming algorithm	Stemming Arabic words with a dictionary	Arabic corpus	Accuracy
[51]	2009	Presents and compares three techniques for the reduction	Height stemming and word clusters	Create dataset	Recall and precision
[52]	2010	Sought to determine the effect of 5 measures with two types of preprocessing for R document clustering	The Information Science Research Institute stemmer	1680 documents	Cosine, Jaccard, Pearson, Euclidean, and DAvg KL
[53]	2010	To create an efficient rule-based light stemmer	Light stemmer for the Arabic language	—	—
[54]	2010	Aim to present a new dictionary-based Arabic stemmer	Local stem	The dataset contains 2966 documents	Accuracy
[55]	2010	Aim to design Arabic morphological analysis tools	Stemming and light stemming	Open-Source Arabic Corpus	Accuracy
[56]	2011	Aim to work with many techniques for ATC	Stop-word removal	2363 documents	Recall and precision
[57]	2011	Improved stemming to extract the stem and root of words	Dictionary-based stemmer	Collected Arabic corpus	Accuracy
[58]	2012	Aim to increase accuracy	3 stemmers	House corpus collected	Accuracy
[59]	2012	Propose the first nonstatistically accurate Arabic lemmatizer algorithm that is suitable for information retrieval (IR) systems	An accurate Arabic root-based lemmatizer for information	The dataset contains 50 documents	Accuracy
[60]	2013	Investigates the relevance of using the roots of words as input features in a sentiment analysis system	Tashaphyne stemmer with ISRI stemmer and Khoja stemmer	Penn Arabic Treebank with movie corpus	Accuracy, recall, precision, and F1-score
[61]	2013	Aim to improve khoja	Enhancement of khoja	House corpus collected	Accuracy
[62]	2014	Aim to design a model for the extraction of the word root	Stemmer for feature selection	CNN FROM OSAC	Recall, precision, and F1-score
[63]	2014	Aim to design a light stemmer	Novel root-based Arabic stemmer	Dataset consists 6081 Arabic words	Accuracy
[64]	2014	Aim to design an analyzer for dialectal Arabic morphology	Analyzer called ADAM	SAMA databases	—
[65]	2014	Aim to compare studies for stemming	Khoja stemmer with chi-square	CNN FROM OSAC	Recall
[66]	2015	To study and compare the effect of three stemmer algorithms	Root extractor, light, and khoja stemmer	Arabic WordNet	F1-score
[67]	2015	To improve stemming P-stemmer	P-stemmer	House corpus collected	F1-score
[68]	2015	Aim to root extraction using transducers and rational kernels	Root extraction	Saudi Press Agency dataset	Accuracy, recall, precision, and F1-score
[69]	2015	To introduce a new stemming technique	Approximate stemming	—	Accuracy and F1-score

Table 2: Continued.

Ref.	Year	Objective	Method	Dataset	Evaluation matrices
[70]	2015	To build a new Arabic light stemmer	A new algorithm for light stemming	The dataset consists of 6225 Arabic words	Accuracy
[71]	2016	To improve accuracy by design feature selection	Normalization and stemming techniques	Dataset 1, dataset 2, and dataset 3) collected from the website https://www.aljazeera.net	Accuracy, recall, precision, and F1-score
[72]	2016	To study the Khoja stemmer and the light stemmer stemming algorithm	Normalization, root base stemming, and light stemming approaches	Create a dataset with 750 documents	Recall, precision
[73]	2016	To design a software tool for AT stemming	Light stemmer	—	—
[74]	2016	Aims to highlight the effect of preprocessing tasks on the efficiency of the Arabic DC system	Stemming techniques with	House corpus collected	F1-score
[75]	2016	Aim to study a fast and accurate segmenter	Arabic segmenter	—	—
[76]	2017	To review stemming ATs	Effective Arabic stemmer	—	—
[77]	2017	To implement a new Arabic light stemmer	Light stemmer	ARASTEM dataset	Using Paice's parameters
[78]	2017	To design a new morphological model based on regular expressions	Morphological model	Some Surat from the Holy Quran	False positive and false negative rate
[79]	2017	Evaluation study among several preprocessing tools in Arabic TC	Among several preprocessing tools	Alj-News Dataset and Alj-Mgz Dataset	F1-score
[80]	2018	To design the FS technique and improve the accuracy	Improved chi-square	Open-Source Arabic Corpora (OSAC) and (CNN)	Precision, recall, and F1-score
[81]	2018	Conduct a comparative study about the impact of stemming algorithms	Stemming	CNN-Arabic site and contains 5070	Recall
[82]	2019	To study different steamer AR Stem, Information Science Research Institute, and Tashaphyne	Stemming	CNN-Arabic site and contains 5071	F1-score
[83]	2019	Aim to extract a root by processing word-stemming levels to remove all additional affixes	Root extraction and stemming	Collection of 350 documents	Accuracy
[84]	2019	Aims to review the state of the retrieval performance of Arabic light stemmers	Light stemmers	TREC data	Accuracy
[85]	2019	To a novel method that detects not only domain-independent stop words	Stop word	Corpus combines 1261 Facebook comments, 781 tweets, and 32 reviews	F1-score
[86]	2020	To discuss the impact of the light stemming algorithm on text classification	Study the effects of the light stemming	BBC Arabic dataset	Recall, precision
[87]	2020	To discuss the impact of a stemming algorithm on word embedding representation	Stemming techniques	ANT version 1.1 and SPA corpus	F1-score
[88]	2021	Design a new method to prepare and analyze the AT	Normalization, such as shape repeated letters, non-normal words, and spelling mistakes	Collect data character	—
[33]	2024	Study how ATC work on hate speech	Many methods	Survey	—

4.4. Normalization. This refers to a collection of many documents of various formats that are transformed into a standard format such as ".txt" in case our data are represented as a multidocument. On the other side, when our data are represented as a single document, the normalization here is to make all words in the same form, and there are many techniques such as stemming. Finally, normalization takes in rules or regular expressions [71].

4.5. Lemmatization. Lemmatization reduces a word to its simplest form by replacing the suffix or prefix of a word with a different one or removing the suffix or prefix from the word utilizing lexical knowledge [90, 91].

4.6. Stemming. Text stemming is the process of reducing inflected or derived words to their common canonical form. For example, 'teacher', 'school', and 'studying' might be reduced to their root forms such as 'teach', 'school', and 'study' (درس, الٰى, يدرس, مدرسه, مدرس) [90]. There are various types of stemming, for example, root-based stemmers—Khoja, light-based stemmers—Larky, and statistical-based stemming like N-grams, as shown in Figure 2.

Larkey and Connel [43] implemented and improved normalization and stemming methods for AT. In addition, they have created a dictionary and expanded inquiries for AT with no prior knowledge of the language. Larkey et al. [44] further developed several light stemmers based on heuristics and statistical stemmers for Arabic retrieval. A morphological stemmer that sought to locate the root for each word proved more successful for cross-language retrieval than the best light stemmer did. Duwairi et al. applied different FS approaches to the Arabic corpus. They compared stemming and light stemming, coming to the conclusion that light stemming improves classification accuracy. Three feature reduction methods based on stemming, light stemming, and word clusters were proposed with K-NN as classifiers [51]. Mohd et al. attempted to describe the influence of several metrics, such as cosine similarity, Jaccard coefficient, Pearson correlation, Euclidean distance, and averaged Kullback–Leibler divergence on document clustering algorithms with two forms of morphology-based preprocessing [52]. Mansour et al. [47] proposed an autoindexing method for IR to create index words for AT documents while applying different grammatical rules to extract stems. Al-Shammari and Lin [48] introduced a novel lemmatization algorithm for AT and argued that lemmatization is a superior word normalization approach to stemming. Al-Shargabi et al. [56] applied different preprocessing methods and compared the performance of SVM, NB, J48, and SMO classifiers performance and concluded that SMO outperformed the other classifiers.

Hadni et al. [58] implemented an effective hybrid approach for ATC that is reported to supersede Larky, Khoja, and N-gram stemmer. Oraby et al. [60] studied the effect of stemming methods on Arabic SA. Their accuracy results were 93.2%, 92.6%, 92.6%, and 92.2% for Tashaphyne, stemmer, ISRI stemmer, and Khoja stemmer, respectively. Bahassine et al. [62] studied the effect of the origin stemmer and Khoja's stemmer on Arabic document classification. CHI statistics were used to reduce the number of selected features. Their proposed stemming method outperformed Khoja's stemmer. Al-Kabi et al. [63] proposed a new light stemmer for AT. The empirical evaluation indicated that the proposed stemmer's accuracy is higher than one of the two well-known Arabic stemmers utilized as a baseline. Salloum and Habash [64] presented an analyzer for dialectal Arabic morphology for AT. It is an analyzer for dialectal Arabic, and its performance is comparable to an Egyptian dialectal morphological analyzer. Yousif et al. [66] presented an ATC system based on NB with a conceptual representation based on Arabic WordNet. They assessed the impact of three stemming algorithms: a light stemmer, a Khoja stemmer, and a best-performing root extractor.

Kanan and Fox [67] developed a taxonomy for Arabic news with automatic classification techniques using binary SVM classifiers and a novel Arabic light stemmer called P-Stemmer. Nehar et al. [68, 71] enhanced ATC utilizing an improved feature set, including the BoW and term-frequency approach and the frequency ratio accumulation method classifier. Nehar et al. provided a new approach to root extraction based on using an Arabic pattern stemmer to classify AT. Nasef and Jakovljević [73] presented the categorization of AT using stemming. The software is based on an open-source version of the Lucene-based light stemmer for Arabic, and it allows for stemming and categorization into 12 classes. Mustafa et al. [76] presented an extensive survey on Arabic stemmer. Abainia et al. [77] suggested the design of a unique Arabic light stemmer based on certain new principles for smartly removing prefixes, suffixes, and infixes. It is also the first book to address the irregular norms of Arabic infixes.

Bahassine et al. [80] increased the accuracy of Arabic document categorization; FS approaches employing IG, MI, and CHI were used. Boukil et al. [81] proposed the classification of Arabic documents while using stemming techniques as FE systems and KNN as a classifier. Alhaj et al. employed various stemmers, including Information Science Research Institute (ISRI), Tashaphyne, and ARLStem for ATC with SVM as the best-performing classifier. They further studied Arabic document classification utilizing light stemming techniques with FE techniques such as BoW and TF–IDF. Moreover, different FS methods, such as CHI, IG, and singular value decomposition (SVD), were used to select the most relevant features [82, 86]. Belal proposed a system for stemming word-level levels to extract a root in the process of removing all additional affixes. Eliminating all further affixes is proposed as a technique for stemming word-level levels to extract a root. If the procedure of matching between a word and proper names is accessible, remove the affixes using patterns and rules based on root dictionaries [83]. Ouahiba and Othman review the performance of various Arabic Light stemmers and conclude light 10 is the outperforming stemmer [84]. Almuzaini and Azmi [87] discussed the effect of Arabic document classification by stemming strategies and word embedding on different DL models, including CNN, CNN–long short-term memory network (LSTM), gated recurrent units (GRU), and

attention-based LSTM which has been investigated with Word2Vec representation algorithm. Al-Shammari and Lin produced a novel method for stemming Arabic documents called educated text stemmer. They used stemming weight as an assessment measure to compare the new method's performance to that of the Khoja stemming algorithm [49]. Ayedh et al. [74] investigated the influence of preprocessing tasks on the efficiency of the Arabic document categorization system. Three-classification approaches are utilized in this study: NB, KNN, and SVM. Al-Kabi [61] highlighted the flaws in the Khoja stemmer and brought about 5% improvements in accuracy by adding missing patterns. Nehar [69] developed a novel stemming approach known as "approximate stemming," which is based on the usage of Arabic patterns using transducers without relying on any dictionary. Aljlayl and Frieder proposed rule-based light stemming and demonstrated its performance better than a root-based algorithm [45]. Kchaou and Kanoun [50] proposed a method for stemming AT that works similarly to Khoja's strategy, but the difference here is that there are two dictionaries, one for roots and another for radicals. It addresses handicapped roots and radicals in Khoja.

Kanan et al. proposed a novel light stemming from AT and demonstrated its effectiveness in improving search in IR Elshammari [53]. Al-Shammari proposed a context-dependent stemmer without relying on a dictionary and improved ATC by utilizing a new free Arabic stemmer dictionary [54]. The proposed stemmer is compared with the root-based and light stemmers and outperforms them. Alhanini and Aziz proposed an improved stemmer for extracting the stem and root of Arabic words to address the shortcomings of light stemming and dictionary-based stemming. However, the proposed stemmer does not address the issue of broken (irregular) plurals [57]. El-Shishtawy [59] proposed a nonstatistical lemmatizer that uses several Arabic knowledge resources to produce accurate lemma forms and relevant features that can be utilized in IR systems. Abdelali et al. [75] proposed a Farasa Arabic segmenter based on SVM ranking with linear kernels that is comparable to the state. Said et al. reviewed several preprocessing tools in ATC and compared the raw text within many techniques, such as Al-Stem stemmer, Sebawai root extractor, and RDI MORPHO3 stemmer [79].

Elghannam [92] created a new technique for identifying the domain of a corpus. The detection is domain-independent and domain-dependent stop words. Othman et al. developed a new framework based on regular expressions and Arabic grammar rules to extract and recognize an Arabic sentence's syntax analysis [78]. Hegazi et al. [88] designed an approach that provides a framework for building effective apps for analyzing and processing AT on social media.

5. Representation and Feature Engineering

ML algorithms cannot understand unstructured text as humans do unless represented in terms of numbers. Hence, text representation is a process of converting unstructured text into its structured equivalent representation, which ML algorithms can understand and interpret. One of the most effective approaches to text representation is word embedding, which captures the semantic and syntactic relationships between words in a continuous vector space. Traditional techniques such as BoW and TF–IDF treat words as discrete entities, failing to capture contextual meaning. Machine-readable representations of text can be constructed using various representation methods. Figure 4 and Table 3 explore a different type of level representation first. Then, different feature extraction techniques are explored.

The basic unit in language is a word, which produces phrases, sentences, and documents. Because of this, word-based representations are the most critical research direction since the total number of words that we can get from any language is huge compared to characters or phrases.

5.1. Representation Based on Character-Level Methods. Character-level representation refers to a way of representing text data where each character in the text is considered a separate unit of analysis, as opposed to word-level or sentence-level representation, where words or entire sentences are treated as units of analysis. Character-level representation is commonly used in NLP tasks such as language modeling, TC, and machine translation. In this approach, each character in a text is mapped to a unique numeric representation using techniques such as one-hot encoding or embedding. One advantage of character-level representation is that it can handle out-of-vocabulary (OOV) words or rare words not present in a predefined vocabulary; each character can be mapped to a unique representation even if it has never been encountered before. However, character-level representation may not capture the semantics of words or phrases and may require more computational resources than word-level or sentence-level representation. The methodology of character-level embedding starts by dividing each Arabic word into basic letter forms and encoding each alphabet separately. There are two ways to represent text at the character level: encoding every alphabet alone or using another technique called N-gram, adding one, two, or three N-grams. The following subsections present the existing work on these representations.

5.1.1. N-Gram Embeddings. N-gram-level embedding divides each Arabic word into basic letter form and encodes each alphabet differently by taking two or three letters. Petasis et al. [138] proposed a model to deal with high dimensionality for ATC using trigram frequency to represent text, and their results demonstrated that trigram text categorization was effective. Al-Thubaity et al. [100] used a neural network to map English vectors from Arabic vectors, develop continuous representations that capture semantic and syntactic features, and test these vectors using intrinsic and extrinsic evaluations. Elghannam et al. [92] proposed a novel bigram character-based method to represent text for a TC system and evaluated it on the Aljazeera News dataset. Mulki et al. [120] proposed a model that uses N-gram embedding for sentiment in many Arabic dialects. Saeed et al. [123] represented text using the N-gram method

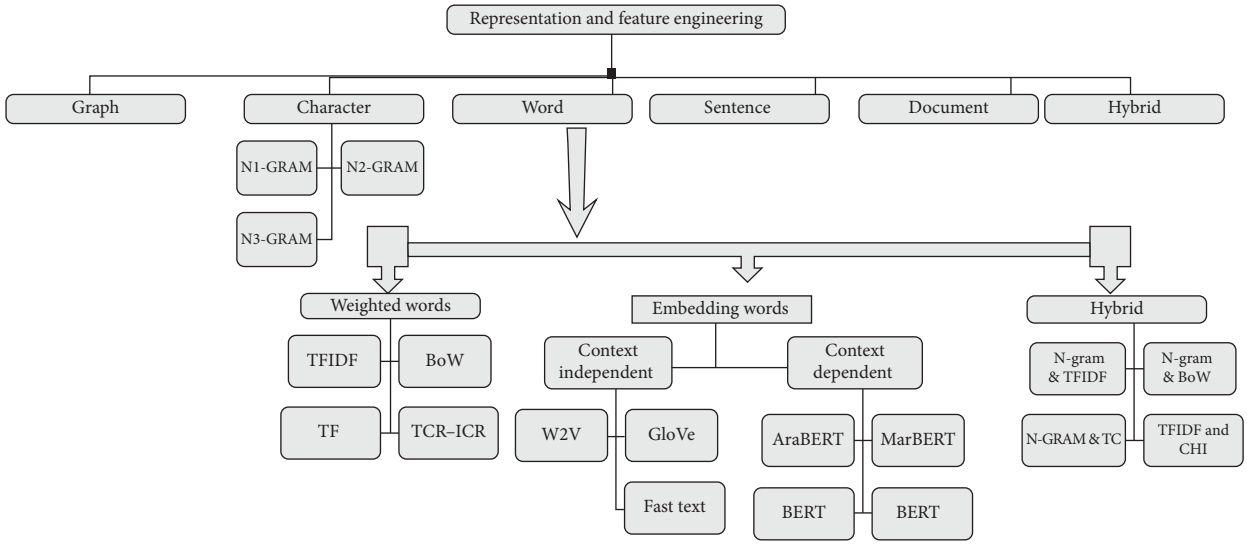

Figure 4: Representation and feature extraction techniques for Arabic text classification.

in numerous classification algorithms to detect spam in Arabic opinion texts, including rule-based and ML algorithms. Elzayady et al. [131] proposed a model for SA by employing CNN for FS and RNN for classification. The method did not address the issue of OOV terms.

5.1.2. Character-Level Embeddings. Character-level embeddings separate each Arabic word into basic letter forms and then encode each alphabet separately. Belinkov et al. represented text at the character level using CNN to distinguish between similar languages and dialects [104]. Ali proposed a CNN-based model to distinguish five dialects of the Arabic language [113]. Omara et al. used a CNN-based model for SA at the character level. Furthermore, the model was evaluated for emotion identification and SA [121].

5.2. Word-Based Embeddings. Word representation refers to the process of encoding words as numeric vectors or embeddings, which can be processed by ML algorithms for various NLP tasks. Word embedding tokenizes a sequence of words at the word level and assigns a vector to each word. In the following section, state-of-the-art word embedding methods have been discussed.

5.2.1. Weighted Words. At the word level, there are many representation techniques to represent text using weighted words, such as TF–IDF. This representation will represent each word and map to several occurrences in the corpus. There are many types as follows:

- BoW: BoW is a feature extraction technique that ignores word order in a text document. Al-Radaideh and Al-Abrat proposed a model based on term weighting for ATC and reduced the number of terms used to generate the classification rules [118]. Alahmadi et al. proposed combining BoW with bag-of-concepts to handle semantic relationships between words. Still, the problem of sparse matrix and complex preprocessing finally did not work with a problem like OOV [98]. Al Sallab et al. proposed three DL models for sentiment classification in AT, each using a different representation method, such as BoW. Their experiments were carried out on the LDC ATB dataset [99]. Alnawas introduced Doc2Vec with ML for SA of AT, and they proposed a continuous vector representation model. They were computed using the PV–DM and PV–DBoW architectures. Furthermore, these vectors were used to train four popular ML methods: LR, SVM, KNN, and RF [126].
- TF–IDF: TF–IDF assigns more weight to fewer common words in a document. Mahmood and Al-Rufaye applied and improved TM by decreasing dimensions utilizing k-means clustering algorithms [110]. Al-Taani et al. proposed an FCM approach to classifying AT by lowering the dimensionality of the representation. They employed SVD for DR, but it has significant disadvantages such as a high complexity time, a high-dimensional space, and a lack of consideration for the semantic level [125].
- TCR–ICF: TCW–ICF is a new method of representation that has been used for ATC. It works like term frequency, which replaces representation based on class instead of a word. Guru et al. proposed TCW–ICF, a novel term weighting system for ATC. Their method improves results by applying DR [114]. Finally, all of their experiments were implemented in the dataset that they created.

5.2.2. Word Embedding. Word embedding converts words to vectors, which can be context-dependent or context-independent. We explore all existing work based on the following:

- Context-Independent Word Embeddings: In this representation, the meaning of surrounding words is ignored; examples include Word2Vec, GloVe, and FastText.

Table 3: Comparative analysis of representation techniques.

Ref.	Year	Objective	Method	Dataset	Evaluation matrices
[93, 94]	2008	Aim to use ML for AT documents classification	Dice measure for classification and representing by trigram frequency statistics	Arabic documents corpus	Precision, recall
[95]	2010	Aim to explore the sentiment of AT at two levels: document and sentence	Design a novel grammatical approach and semantic orientation of words, documents, and sentences at the document and sentence level	44 documents	Accuracy
[56]	2011	To make a comparison of different text classification algorithms	Stop-word removal	2363 documents	Recall and precision
[96]	2012	Propose a conceptual representation for AT representation	Chi-square	Corpus of Arabic texts built by Mesleh	Precision, recall, and F1-score
[97]	2013	Aim to represent AT using rich semantic graph	Graph	A small dataset that contains three paragraphs	—
[98]	2014	Aim to design an algorithm by combining bag-of-words and the bag-of-concepts	TF and TF–IDF	Arabic 1445 dataset and Saudi newspapers (SNP) dataset	Accuracy, recall, precision, and F1-score
[99]	2015	Aim to propose four models for text sentiment classification in Arabic	Bag-of-words word embeddings	LDC ATB dataset	F1-score
[100]	2015	Aim to explore the efficient of word N-grams	N-grams	Saudi Press Agency dataset	Accuracy
[6, 101]	2015	Aim to represent a word in a vector and minimizing for cosine error outperforms	Word embeddings CBOW, SKIP-G, GloVe	Collected ATs	Root mean square error and Pearson's correlation
[102]	2016	To use cosine similarity for ATC	Latent semantic indexing (LSI)	4000 documents on 10 topics	Accuracy
[7]	2016	Aim to solve binary classifiers and detect subjectivity	Word embeddings	Collect datasets to create word representations	Accuracy
[103]	2016	Aim to study sentiment polarity from the AT	Word embeddings Word2Vec	3.4 billion-word corpus.	Accuracy
[104]	2016	Aim to explore the character level for discriminating between similar languages and dialects	Character-level	DSL 2016 shared task	Accuracy and F1-score
[105]	2017	Aim to design a new graph-based algorithm for ATC	Graph	Essex Arabic summaries corpus	Recall, precision, and F1-score
[106]	2016	Aim to prove document embeddings better than text preprocessing methods	Word vectors and Doc2Vec model	BBC, CNN, OSAC, and Arabic Newswire LDC	Precision, recall, and F1-score
[107]	2017	Aim to propose pretrained word representation for AT	Word embeddings (AraVec)	Different resources: Wikipedia, Twitter, and Common Crawl webpages (word embedding)	None
[108]	2017	Aim to use various models for word representations to classify AT	(CBOW, Skip-Gram, and GloVe)	Two datasets: SemEval 2017 and ASTD	F1-score
[109]	2017	Aim to work on three problems for Arabic sentiment analysis	Word embedding with Word2Vec	Syria Tweets dataset	Accuracy recall, precision, and F1-score
[110]	2017	Aim to propose a study that minimizes the high dimension	TF–IDF	Corpus of sport news	Precision, recall, and F-measure

Table 3: Continued.

Ref.	Year	Objective	Method	Dataset	Evaluation matrices
[111]	2018	Aim to utilize deep learning for Arabic sentences classification	Word embeddings	Essex Arabic summaries corpus (EASC)	None
[112]	2018	Aim to design graph model for document	Graph	Arabic dataset	Precision, recall, and F1-score
[113]	2018	Aim to distinguish the 5 dialects using char-level representation	Character level	ADI dataset for the shared task	Accuracy and F1-score
[114]	2018	Aim to propose a new representation technique	TCR–ICF	Collect a new dataset	Accuracy
[115]	2018	Aim to study of several word embedding models is conducted, including GloVe, CBOW, and Skip-gram	GloVe and Word2Vec	Many datasets such as OSAC, LABR, and Abu El-Khair corpus	—
[116]	2018	Aim to compare pretrained vectors of the word for AT	Word embedding (WE) models	Collected from Twitter	Accuracy of 93.5% with AraFT
[117]	2018	Aim to use word representation for sentiment analysis	Word2Vec	Language Health Sentiment Dataset	Accuracy
[118]	2018	Aim to use term weighting and multiple reducts	Term weighting	2700 documents for 9 classes	Recall, precision, and F1-score
[119]	2019	Aim to create word embedding models	ARWORDVEC models	ASTD and ARASENTI	Accuracy and F1-score
[92]	2019	Aim to create a new bigram alphabet approach	Bigram alphabet	Arabic dataset Aljazeera News.	Accuracy
[120]	2019	Aim to introduce N-gram embeddings	N-gram embeddings	Using many western and eastern Arabic datasets	Accuracy, precision, recall, and F1-score
[121]	2019	Aim to study word embedding for text representation	Char level	Merge many datasets	Accuracy
[122]	2019	Aim to design an algorithm for a combined document embedding representation	Word sense	OSAC	Precision, recall, and F1-score
[123]	2019	Aim to propose a new representation model based on N-gram	N-gram	DOSC and HARD datasets	Accuracy, precision, recall, and F1-score
[124]	2019	Aim to introduce a graph-based semantic representation model	Graph	ArbTED	Accuracy precision, recall, and F1-score
[125]	2020	Aim to find a technique for the proposed technique by reducing the high dimensionality	TF–IDF	CNN dataset and Alj-News5 dataset	Precision, recall, and F1-score
[126]	2020	Aim to introduce Doc2Vec and machine learning approaches	PV–DM and PV–DBOW	Five Arabic datasets	Accuracy and F1-score
[127]	2020	Aim to use transfer learning as a new technique for representation	BERT	HARD; ASTD; ArSenTD-Lev; LABR: AJGT	Accuracy and F1-score
[128]	2020	Aim to create embeddings vector based on word and character	Character and word embeddings	TASK	Pearson correlation coefficient
[121]	2020	Aim to apply transfer learning for emotion analysis in Arabic	Character-level representation	Hotel reviews AND 1012 tweets	Accuracy
[129]	2020	Aim to study the impact of BERT model AT formal and unformal	BERT	CREATE TWO	F1-score

Table 3: Continued.

Ref.	Year	Objective	Method	Dataset	Evaluation matrices
[130]	2020	Aim study word-level representations to tackle the Romanized alphabet of Tunisian	Word2vec	Word	Accuracy-measure
[131]	2020	Aim to study Arabic opinion mining using a different type of representation	Unigram, bigram, and trigram	HTL and LABR datasets	Accuracy
[132]	2020	Aim to use pretrained word embedding for Arabic sentiment	ARAVEC and FastText library	Arabic Gold Standard Twitter Data for sentiment analysis (ASTD)	ROC curve
[133]	2020	Aim to classify text utilizing fine-tuned Word2Vec	Word2Vec	Movie review dataset	Accuracy
[134]	2021	Aim to represent text at the word level and investigate an efficient bidirectional LST for classification	Word embedding	ASTD ArTwitter LABR MPQA	Precision, recall, and F1-score
[135]	2021	Aim to introduce a contextual semantic embedding representation	BERT	OSAC	Accuracy and F1-score
[136]	2020	Aim to propose a model for representation embeddings at the different levels	Character, word, and sentence embedding	IMDB movie dataset	Accuracy, precision, recall, and F1-score
[137]	2024	This work combines the trained Arabic language model ARABERT with the potential of long short-term memory (LSTM)	ARABERT	4071 Arabic audio clips	Accuracy, word error rate, character error rate, BLEU score, and perplexity

- Word2Vec: In 2013, Mikolov et al. from Google implemented the W2V model. This model has two hidden layers, a continuous BoW and the second one, Skip-Gram, which both work on a high-dimensional vector for each word. Some researchers have used the following methods for representation.
- Altowayan represented text and created embedded words for SA tasks to represent AT. The embedding of features for binary classifiers was used to detect standard and dialectal AT, and they also presented word embedding as an alternative to extract features for Arabic sentiment classification. Their method depends on word embedding as AT as the primary source of characteristics. Two types of AT have been detected using this representation [7]. Dahou et al. detected Arabic review sentiment polarity and social media from AT. They used to study corpora from two domains: reviews and tweets [103]. Soliman et al. introduced a pretrained distributed representation called AraVec. They make this work open source to support the researcher community. Their model handles syntactic and semantic relations among words [107]. Al-Azani and El-Alfy designed a model for SA to solve three problems: microblogging data, handling imbalanced classes, and addressing dialectical Arabic. The oversampling technique solved the imbalanced dataset problem [109]. Sagheer and Sukkar resented classifying Arabic sentences using CNN models with a representation embedding layer. They have used AraVec as a pretrained system [111]. Alwehaibi et al. implemented SA for AT using the LSTM model on Arabic tweets. They assess the impact of pretrained vectors for numerical word representations that are already available. The experimental findings suggest that the LSTM–RNN model produces acceptable results [116]. Alayba et al. described how they have constructed Word2Vec models from a large Arabic corpus obtained from 10 newspapers in different Arab countries. Different ML algorithms and CNN with various FS methods were applied to the health sentiment dataset. They increase the accuracy of the form from 91% to 95% [117]. Fouad et al. showed that effective word embedding in ArWordVec was developed from Arabic tweets. They created a new approach for detecting word similarity. The experimental results suggested that the ArWordVec models outperform previously available models on Arabic Twitter data. Finally, they applied various models to obtain word embeddings, such as the CBoW, SG, and GloVe methods [119]. Abir Messaoudi et al. presented different word representations of different DL models (CNN and BiLSTM), without using any preprocessing step. They proved that CNN with M-BERT reached the best results compared to others [130]. Sharma et al. proposed a model to perfectly clean the data and generate word vectors from the pretrained Word2Vec model [133]. Elfaik and Nfaoui (2021) proposed a model for ATC. They represented text at the word level and investigated BiLSTM to improve the SA of AT. The F1 measure was 79.41 in LABR datasets. The complexity of preprocessing and time was greater. They did not use character level, which may solve some problems for the Arabic language [134].
- GloVe: It is an unsupervised learning algorithm for obtaining vector representations for words, a strong representation to represent text [90]. The approach is similar to the Word2Vec method. M. A. Z. et al. investigated the effective representation of N-grams as features for ATC. Their experiment used the SPA dataset [101]. Gridach et al. implemented various word representation models, such as CBoW, Skip-Gram, and GloVe utilizing two datasets called ASTD, and SemEval [108]. Suleiman and Awajan studied various word embeddings to represent AT. These techniques are GloVe and Word2Vec. Finally, they conclude that Word2Vec outperforms others [115].
- FastText: Facebook's AI Research Lab released a novel technique to solve the representation issue by introducing a new word embedding method called FastText. Each word is represented as a bag of character N-gram. For example, given the word "محمد" and $n = 4$, FastText will produce the following representation composed of character trigrams: < مح, محم, حمد, مد >. Ibrahim Kaibi introduced NuSVC classifiers to classify AT using word embeddings representations known as AraVec and FastText. They combined both representation models based on the concatenation of their vectors. Evaluate the model using accuracy metrics [132].
- Context-Dependent: It is one type of representation in which the meaning of the context is included. This representation depends on the context of the sentence, which means there will be more simulation of humans.
- AraBERT: Antoun et al. implemented new transfer learning to classify AT. This model called AraBERT achieves the same BERT in English text. They compare multilingual BERT with AraBERT [127]. Chowdhury et al. studied the effects of the BERT model on a mixture of formal and informal texts. They applied new Arabic transfer learning for short-text datasets. They prove that greater generalization was made by the former when compared to others [129]. F. Zahra El-Alami et al. presented embedding representations that handle semantic context to improve ATC. This type of representation solves many complex problems. They implemented and compared their work with AraBERT [135].
- MarBERT: Abdul-Mageed et al. presented two powerful Transformer-based models, especially for Arabic. They train their models on large-to-massive datasets that cover different domains [139].

5.3. Document-Level Methods. Mahdaouy et al. introduced a classification system to classify text and documents in vector space, and their representations for the document in

an unsupervised method are to carry implicit relationships and semantics between words [106].

5.4. Sentence-Level Methods. A sentence representation is usually used in many tasks in natural language. Sentence representation aims to encode the semantic information of the whole sentence into a real-valued representation vector, which could improve the understanding of the context of the text. Farra et al. examined sentiment text for Arabic at two-level document and word. They conclude that the work, which has been done in Arabic, is still limited. They studied a novel grammatical method and the semantic orientation of words with their corresponding [95].

5.5. Representation Based on Hybrid Methods. Hybrid methods try to merge more than one method for text representation, by utilizing some advantage in one method and another advantage from another. Al-Anzi et al. proposed TC for AT and compared some of them. They employed SVD to decrease the dimension and reduce the number of features [102]. El-Alami et al. presented a method that works with two phases of document embedding and sense disambiguation to improve accuracy. They implemented several experiments on the Open-Source Arabic Corpora dataset. However, there are some limitations, such as using TF–IDF representation, which takes a sparse matrix representation, a complex preprocessing, especially using the Khoja stemmer, and using a lexicon will cover only some vocabulary, so these cannot be appropriate for the Arabic language since it has a rich vocabulary and rare words [122]. Alharbi et al. designed a model to classify microblogs on social media using word and character representation. At the same time, they presented a new technique that joins different levels of word embedding [128]. El-Affendi et al. developed a novel DL multilevel model that uses a simple positional binary embedding scheme to compute contextualized embedding at the character, word, and sentence levels simultaneously. The suggested model is also shown to generate new state-of-the-art accuracies for two multidomain problems [136].

5.6. Representation Based on Graph Methods. The representation of text as a graph is one of the essential preprocessing steps in data and TM in many domains, such as TC. The graph representation approach is used to represent text documents in a graph to handle text features such as semantics [124, 140]. El Bazzi et al. implemented a system to classify documents using a graph model for representation. They studied the impact of the semantic relation between the text tokens on the papers [112]. Ismail et al. presented a system to summarize and classify AT using a rich semantic graph (RSG). It is a suitable method that supports the development of the Arabic language [97]. Hadni and Gouiouez proposed a new graph approach for representing text and classifying AT. This is accomplished through using BabelNet knowledge [105]. Etaiwi and Awaja introduced a graph representation to classify AT. Their model was evaluated using different metrics such as precision, accuracy, recall, and F1-score [124].

6. DR

Representation of text in vector space models (VSMs) such as BOW has several limitations, for example, sparse matrices. These methods are pretty expensive in terms of time complexity and memory utilization. Many researchers utilized DR to limit the size of the feature space to address this limitation. Existing DR methods used in AT categorization are discussed in this section and shown in Figure 5 and Table 4.

6.1. Feature Selection. In general, FS has three categories known as embedded, wrapper, and filtering techniques, but in TC, the processes are more likely to use filters due to many features. Mesleh applied a TC system using SVM with CHI; simultaneously, they suggested other FS algorithms for future work [141]. Mesleh et al. collected a house dataset and used six FS techniques for ATC purposes. Based on their different experiences, they noted that FS is beneficial in increasing the accuracy of ATC [142]. Duwairi discussed three feature reduction approaches to improve accuracy in AT. At the same time, they made comparisons for stemming, light stemming, and word clustering [51]. Bahassine et al. developed a new method for the classification system of AT and applied CHI to improve classification accuracy and decrease size [80]. Larabi Marie-Sainte and Alalyani implemented SVM and FS methods used in numerous scenarios to study the classification of AT. Due to the complexity of Arabic, it was not done intensively. Their experimentation was evaluated using various matrices like precision, recall, and F1-score [150]. Rashid et al. implemented FS to increase the accuracy of ATCS, while precision, recall, and F1-score were used [151]. Belazzoug et al. proved that FS is important in enhancing the ATC system. They used BoW for representation; the main problem was having many features [152].

6.2. Feature Extraction. Mohamed applied a new algorithm for extracting features and decreasing the dimension. Principal component analysis (PCA), non-negative matrix factorization (NMF), and SVD have been used for clustering approaches. Finally, he evaluated three well-known techniques to demonstrate the advantages and disadvantages of each [153].

6.3. Optimization. Only a few works have been explored for ATC when compared to existing techniques. Chantar et al. designed a new method for FS to improve TC called the grey wolf optimizer (GWO). This method is a wrapper-based FS [157].

6.4. Hybrid. Sabbah and Selamat presented a hybrid FS method to improve the TC system. They represent text-using TF–IDF to represent text. At the same time, they used other

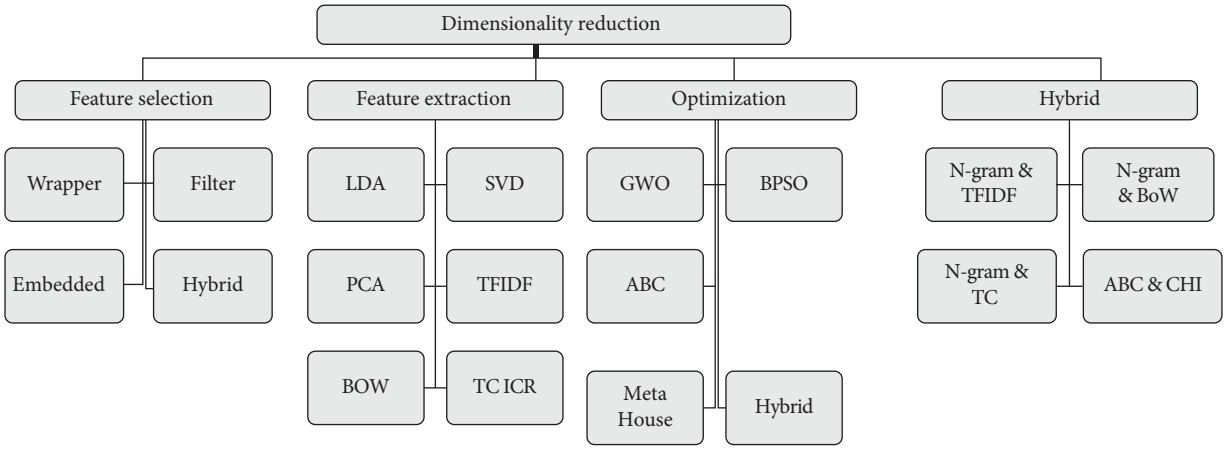

Figure 5: Dimensionality reduction techniques for Arabic text classification.

techniques, such as PCA, to decrease dimension [155]. Hijazi et al. created a novel FS technique that combines artificial bee colony (ABC) and CHI. CHI has three advantages: quick and easy to use. The second phase used the ABC [156]. Chantar et al. proposed an ATC system using KNN and SVM to classify text and binary particle swarm optimization (BPSO), hybridized to select features [157]. Thabtah et al. introduced TC using the NB algorithm based on the CHI feature selection method. They have used many metrics for evaluation, such as F macro, recall, and precision [143].

Chantar et al. proposed an ATC system using KNN and SVM with BPSO as FS [157]. Hussein and Awadalla presented a TC system using different classification algorithms. By combining synonyms, dimensionality has been utilized as a semantic feature selection method [144]. Karima et al. proposed a conceptual representation of ATR. We used AWN to map the terms to the concept [96]. Zrigui et al. presented a conceptual representation for working with ATC. At the same time, AWN maps terms to the concept [145]. Saad et al. developed a new strategy for reducing the number of features by merging semantic synonyms and enhancing ATC [158]. Zaki et al. proposed an Arabic document system based on traditional models. Simultaneously, N-grams with TF–IDF representation techniques were applied [147]. Abu-Errub implemented TF–IDF representation techniques to classify documents into the right class. At the same time, they used the CHI method for FS [148].

7. Classification Models

Once the representation and choosing the optimal feature have been done for a given text through an optimal representation technique, the selection of such a classifier is a crucial task in ATC [15]. Many classification algorithms have been implemented in the literature on ATC shown in Figure 6 and Table 5. One of the significant challenges in applying ML to low-resource languages such as Arabic is the limited availability of high-quality labeled datasets. Unlike widely studied languages such as English, Arabic suffers from data scarcity, particularly in specialized domains. Furthermore, the complexity of Arabic morphology, including rich inflection, derivation, and agglutination, poses additional difficulties in feature extraction and representation. Dialectal variations across different regions further complicate text classification, as models trained on modern standard Arabic (MSA) may struggle to generalize across various dialects. Additionally, the lack of standardized preprocessing techniques and annotated corpora makes it challenging to fine-tune models effectively. Addressing these issues requires the development of transfer learning approaches, data augmentation techniques, and hybrid models that can leverage both supervised and unsupervised learning methods to enhance performance in low-resource NLP tasks.

7.1. Rule-Based (Lexicon or Dictionary). Rule-based classifiers are one type of classifier that makes class decisions based on various "if. . .else" rules. Because these rules are simple to understand, these classifiers are commonly used to generate descriptive models. The condition used with "if" is referred to as the antecedent, and the predicted class for each rule is referred to as the consequent. Rule-based SA refers to the study conducted by language experts. The outcome of this study is a set of rules (lexicon or sentiment lexicon) according to which the words classified are either positive or negative. A dictionary-based (lexicon-based) SA uses lists of words called lexicons. In these lists, the words have been prescored for sentiment.

Different methods have been used under rule-based approaches, such as lexicons and dictionaries. ATCs systems use these rules with string comparisons of text for some tasks. A few researchers have used this method. Nwesri et al. introduced various algorithms to specify foreign words utilizing lexicons, patterns, and N-grams, and they have proven that the lexicon approach was the best [94]. Thabtah et al. conducted in-depth research on the problem of ATC and evaluated the efficacy of different rule-based classification algorithms [164].

Table 4: Comparative analysis of dimensionality reduction technique.

Ref.	Year	Objective	Method	Dataset	Evaluation matrices
[141]	2007	Aims to implement an SVM with chi-square	Chi-square	Arabic data	Precision, recall, and F1-score
[142]	2007	Aims to explore the effectiveness of different feature selection methods	Chi-square	Arabic data	Precision, recall, and F1-score
[51]	2008	Aim to introduce three feature reduction techniques and compare them	Cluster with stemming	15,000 documents	Precision and recall
[143]	2009	Aims to study the impact of the NB algorithm with the chi-square	Chi-square	SPA	Recall, precision, and F1-score
[144]	2011	Aim to study a feature reduction algorithm	Feature selection synonyms merge	House Arabic documents	F1-score
[96]	2012	Propose a conceptual representation for AT representation	Chi-square	Corpus of Arabic texts built by Mesleh	Precision, recall, and F1-score
[145]	2012	Aim to introduce LDA (latent Dirichlet allocation) algorithm b	LDA (latent Dirichlet allocation)	House corpus of ATs	F1-score
[146]	2013	This thesis introduces a new algorithm for feature selection called binary particle swarm optimization	The feature selection process, the filter wrapper approach	Akhbar-Alkhaleej, Arabic Alwatan, Al-Jazeera-News Arabic	Recall, precision, and F1-score
[147]	2014	Aim to improve the AT categorization system by reducing the dimension	Radial basis function	House Arabic documents	Precision, recall
[148]	2014	Proposes a new method for ATC in which a document is compared with predefined documents, using the chi-square measure	TF–IDF and chi-square	House containing 1090 documents	—
[101]	2015	Aim to improve accuracy by representing a word and decreasing the cosine error	Word embeddings CBOW, SKIP-G, GloVe	Collect home data	—
[106]	2016	Aim to prove that representation is better than text preprocessing method	Word vectors and Doc2Vec	BBC, CNN OSAC corpora2, Arabic Newswire LDC	Precision, recall, and F1-score
[110]	2017	Aim to propose a study that minimizes the features	TF–IDF	200 sports news corpus	Precision, recall, and F1-score
[80]	2018	Aim to improve the chi-square	Improve chi	Open-Source Arabic Corpus (OSAC)	Precision, recall, and F-measure
[149]	2018	Aim to investigate one of the most successful classification algorithms which are C4.5.	Chi-square and symmetric uncertainty	Arabic dataset	Precision, recall

Table 4: Continued.

Ref.	Year	Objective	Method	Dataset	Evaluation matrices
[150]	2018	Aim to propose a new feature selection method	Feature selection	Open-Source Arabic Corpus (OSAC)	Precision, recall, and F1-score
[151]	2019	The proposed feature selection approach improves the accuracy	Feature selection	—	Precision, recall, and F1-score
[152]	2019	Propose a solution for the main problem, a large number of involved features	Feature selection	—	—
[153]	2019	Aim to compare three-dimensional reduction methods	PCA SVD NMF	2 linguistic corpora for English and Arabic	—
[154]	2019	Aim to design a method for feature selection	Feature selection	NN, BBC, and OSAC	—
[155]	2019	Aim to introduce hybridization feature set methods	Hybridized feature set	Dark Web Forum Portal	F1-score and accuracy
[156]	2020	Aim to improve the feature selection method by merging the chi-square and artificial bee colony	Hybrid	BBC	F1-score
[157]	2020	Aim to improve and enhance the wrapper FS called the binary grey wolf optimizer	Grey wolf optimizer	Alwatan, Akhbar-Alkhaleej, and Al-Jazeera-News	Precision, recall, and F1-score
[158]	2012	Aim to strengthen AT categorization system utilizing feature selection	Synonyms merge technique	House Arabic documents	F1-score

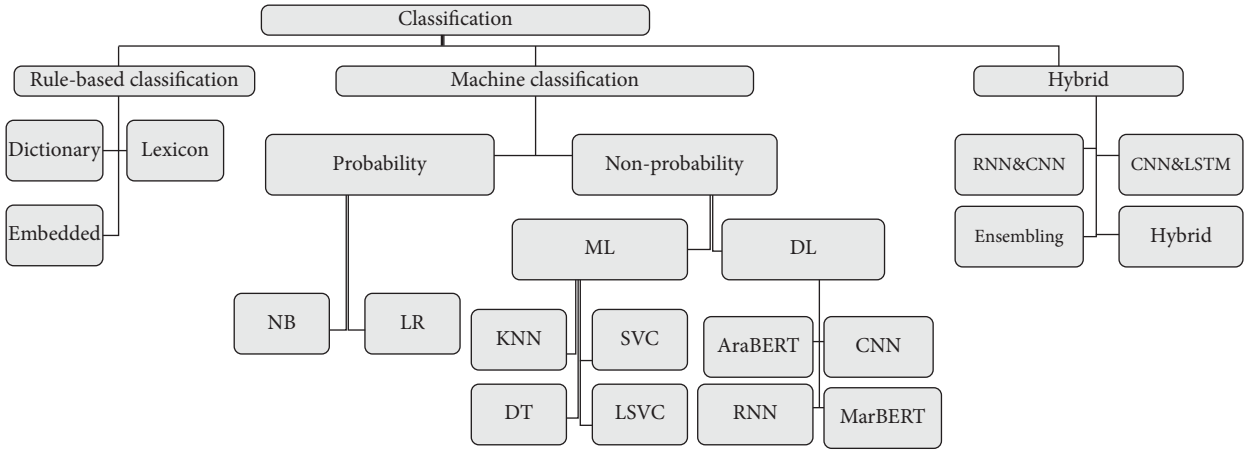

Figure 6: Classification technique for Arabic text.

7.2. Classification Using ML Algorithm. ML and DL approaches achieve state-of-the-art results on ATC. In this section, we explore the related work regarding ATC.

7.2.1. Probability. El Kourdi et al. studied a statistical ML algorithm based on NB to classify nonvocalized AT. The NB categorizer is evaluated using cross-validation trials [159]. Yousif et al. applied NB to classify texts utilizing WordNet for representation and different stemmers to compare them [66]. Syiam et al. presented a Rocchio classifier algorithm for TC, which outperformed KNN. At the same time, they are addressed by combining DR techniques such as stemming and FS to reduce the cost classification process [160].

7.2.2. Nonprobability

- Traditional ML: Al-Harbi et al. implemented AT documents on seven corpora generated for AT using a recognized statistical technique. Their method improved performance by utilizing FS and SVM with C5.0, which has been used. Finally, they conclude that C5.0 provides superior accuracy [161]. Mohammad et al. used a polynomial neural network in TC to produce successful outcomes [177]. Harrag and El-Qawasmah built a neural network for ATC and singular value decomposition to improve accuracy and reduce error [22]. Thabtah et al. studied different representation methods, such as term weighting approaches with the KNN algorithm for classification. In their comparison, they used the F1 evaluation metric [162]. El-Halees studied and combined approaches to classifying Arabic documents. He used three methods in the sequence: first, lexicon, ME, and k-NN to classify AT in different steps [163].
- DL: Gridach introduced a new architecture that represents text at the character level and word level to name the entity recognition. The problem of vanishing gradients arises in the context of long sequences, particularly in tasks like text classification, making it difficult for models to learn long-range dependencies. The OOV problem is still there because word-level embedding cannot predict new words that have not been seen before [168]. Abu Kwaik et al. investigated the DL technique to detect dialectal AT. Their architecture was word-level representations. The experimental results had an accuracy of 81% in the LABR dataset and 85.58% in the ASTD dataset [174]. Abuhaiba and Dawoud proposed combining rules, followed by two classification stages for ATC [169]. Gridach et al. proposed a DL system for SA using DL and CBoW, Skip-Gram, and GloVe for representation [108]. Alayba et al. combined CNN and LSTM networks for Arabic sentiment categorization. Because of the complexity of Arabic morphology and orthography, it also investigated the usefulness of applying various levels of SA. Abdullah et al. described a system to detect and classify Arabic tweets utilizing word and document embeddings. They used a combination of CNN–LSTM for the classification task [172]. Elnagar et al. used Word2Vec embeddings trained on the Wikipedia corpus for text classification. They report the accuracy of 91.18% achieved by convolutional GRU on the SANAD corpus. However, applying normalization by replacing the letters (ÎÌÍ) with a letter (ا) in some cases will change the meaning; for example, فأر (means "mouse") will transform to "فار" (means "escaped") [175]. Finally, their works are based on filtering all alphabets and deciding whether they belong to Arabic. They eliminated non-Arabic alphabets, which added confusion when we had the text from other languages like Urdu. Abu Kwaik et al. proposed a new model for TC by a combination of LSTM–CNN to detect the dialectal of AT [174]. Daif et al. presented the DL structure for AT document classification using image-based characters. Each Arabic character or alphabet was represented as a 2D image. They trained their model from start to finish with the weighted class loss function to avoid the imbalance issue. They produced AWT and APD datasets to evaluate their model [179]. El-Alami et al. proposed an AT categorization method based on bag-of-concepts and deep

Table 5: Comparative analysis of classification techniques.

Ref.	Year	Objective	Method	Dataset	Evaluation matrices
[159]	2004	Aim to apply Arabic web documents classification using NB	Naïve Bayes	Collected 300 web documents per category	Accuracy
[160]	2006	Aim to present a system for ATC	Rocchio classifier	Collected data corpus	—
[94]	2006	Aim to identify foreign words using three-classification method	Lexicons for AT	Collected dataset	—
[13]	2007	Aim to apply three algorithms for AT text classification techniques	KNN, Rocchio, and Naïve Bayes	1445 document	Accuracy, precision, recall, and F1-score
[161]	2008	An implementation classification using a recognized statistics technique	SVM	Different	Accuracy
[162]	2008	This paper investigated different vector space models and use the KNN algorithm	SVM	Collected	F1
[22]	2009	Aim to classify Arabic documents using artificial neural network	SVD and neural networks	Hadith! corpus	Accuracy precision, recall, and F1-score
[163]	2011	Proposed to classify documents using lexicon and k-NN	K-nearest	NONE	Precision, recall, and F1-score
[164]	2012	Aim to apply different rule-based classification algorithms	Rule-based, DT (C4.5), rule induction (RIPPER), hybrid	Published corpus	Rule-based
[165]	2012	Aim to compare six well-known classifiers after applying feature selection.	Naive Bayes without fs and maximum entropy with information gain	Arabic datasets	Precision, recall, and F1-score
[146]	2013	Aim to apply feature selection to improve accuracy	The feature selection process, the filter wrapper approach	Akhbar-Alkhaleej, Arabic Alwatan, Al-Jazeera-News Arabic dataset	Precision, recall, and F1-score
[166]	2014	Aim to improve accuracy by using a different classification algorithm	SVM, NB, and C4.5	Using Arabic Wikipedia	Precision, recall, and F1-score
[167]	2014	Implemented the key nearest neighbor (KNN) algorithm	KNN	Dataset contains 621 documents	Precision and recall
[66]	2015	An implementation of a Naive Bayesian classifier for classification	Naive Bayesian classifier	BBC Arabic corpus	—
[9]	2016	Aim to classify text using a graph-based approach	KNN, Rocchio, and Naïve Bayes algorithms	Corpus of 1084 documents	F1-score
[168]	2016	Aim to classify AT utilizing a hybrid method	Conditional random field and LSTM	NONE	Precision, recall, and F1-score
[169]	2017	Aim to classify AT documents using a different algorithm	Rules, NB, LR, and AdaBoost with bagging	CNN BBC OSAC	Accuracy
[108]	2017	Aim to use DL for sentiment analysis	CBOW, Skip-Gram, and GloVe	ASTD and SemEval 2017 datasets.	F1-score
[170]	2017	Aim to use neural networks and SVM and compare them	RNN	HOTEL DATA	Accuracy and F1-score
[8]	2018	Aim to implement convolutional neural network (CNN) to classify AT from large datasets	CNN	Large dataset collection	Accuracy
[171]	2018	Aim to use a combination of CNNs and LSTMs	CNN–LSTM	Arabic health services (AHS) dataset	Accuracy
[172]	2018	Aim to design architectures to improve accuracy	CNN–LSTM	Task 1's datasets	Accuracy
[173]	2018	Aim to classify text using different classification techniques	KNN, and Naïve Bayes algorithms .svm	CNN dataset	Precision, recall, and F1-score
[174]	2019	Aim to combine LSTM with CNN	LSTM with CNN	LABR, ASTD	Accuracy
[175]	2019	Aim to classify documents using a convolutional GRU	Many models	Khaleej Arabia akbarona	Accuracy

Table 5: Continued.

Ref.	Year	Objective	Method	Dataset	Evaluation matrices
[176]	2019	Aim to classify Hadith document using different DT, RF, and NB	DT, RF, and Naïve Bayes	Hadith DATA	Accuracy
[174]	2019	Aim to detect dialectal Arabic using deep learning	LSTM, CNN	LABR, ASTD	Accuracy
[177]	2019	Aim to classify text using polynomial neural network	Polynomial neural networks	Arabic dataset	Precision, recall, and F1-score
[178]	2019	Aim to classify text utilizing the narrow structure of CNN	Narrow convolutional neural network	Twitter datasets for dialect	Accuracy, precision F1-score
[179]	2020	Aim to represent text as an image-based character to classify a document	CNN1D	They have created AWT and APD	F1-score
[180]	2020	Aim to classify text based on deep auto encoder representations and bag-of-concepts	A deep Autoencoder classifier	OSAC	Precision, recall, and F1-score
[181]	2020	Aim to classify AT documents by a combination of CNN and RNN	CNN and RNN	OSAC	Precision, recall, and F1-score
[182]	2020	Aim to use CNN, LSTM, and their combination for classification	CNN and LSTM	OSAC	F1-score
[183]	2020	Proposed methods to achieve very high accuracy using CNN	CNN	15 different	Accuracy
[184]	2020	Aim to use the CNN architecture with LSTM to classify AT	CNN	LABR ASTD ArTwitter	Precision, recall, F1-score, and accuracy
[185]	2021	Aim to compare four machine learning algorithms in the task of ATC	Artificial neural network, DT, and LR	AJGT, ASTD, Twitter	Precision, recall, F1-score, and accuracy
[186]	2021	Aim to classify AT utilizing two models, GRU and IAN-BGRU	SVM, KNN, J48, and DT based on gated recurrent units and an interactive attention network based on bidirectional GRU	Arabic hotel reviews dataset	Precision, recall, F1-score, and ROC (%)

Autoencoder representations to eliminate problems like explicit knowledge in semantic vocabularies using Arabic WordNet. Their method combines implicit and explicit semantics and reduces feature space dimensionality. They achieved the best results by 94% and 93% for precision and F-measure, respectively. However, their methods still suffer from the complexity of preprocessing and they cannot properly handle the level of vocabulary. Finally, they do not handle the Arabic language ambiguity issue and enhance their system's performance by utilizing sense embedding techniques [123, 180]. Ameur et al. proposed a combination of CNN and RNN for AT document categorization using static, dynamic, fine-tuned, and word embedding. The DL CNN model automatically learns the most meaningful representations from Arabic word embedding space. They evaluated their proposed DL model using the OSAc dataset. By comparing the performance with the individual models of CNN and RNN, their proposed hybridization model helped improve ATC's overall performance. There are some limitations in, such as normalization by changing some alphabet to another form, but in some cases, the meaning will change; for example, "كرة" (means football) will transform to "كره" (means hate) [181]. El-Alami et al. studied a hybrid of DL (CNNs and LSTM) that shows promise for huge datasets. They resolved issues such as the polysemous term. Simultaneously, a method for context meaning employing embedding and word sense disambiguation was proposed [182]. Alhawarat and Aseeri suggested the CNN model for ATC, but it takes a long time to train compared to ML approaches. They produced good results utilizing 15 freely available datasets [183]. Ombabi et al. suggested a DL model for Arabic SA, with this model fully combining a one-layer CNN architecture with two LSTM layers. As the input layer, this approach is handled by word embedding and FastText [184]. Al-Smadi et al. proposed an SVM approach that outperforms the other RNN approach on Arabic hotels' reviews [170]. Alali et al. suggested that CNN utilizes representations to classify tweets. A sensitivity study was carried out to assess the influence of different combinations of structural features [178].

7.3. Hybrid. This approach combines more than one method for text classification; for example, it combines the rule-based and ML algorithms to achieve the maximum possible effectiveness. Kanaan et al. demonstrated many classification algorithms for classifying AT. They used NB, KNN, and Rocchio. The NB was the most effective [13]. Alahmadi et al. proposed a categorization system for AT utilizing a hybrid technique. They employed BoW and BoC for representation to tackle the semantic problem. However, the issue of sparse matrix and complexity with preprocessing finally did not work with the problem of OOV [166]. Bazzi et al. proposed a classification system using graph-based representation. First, a graph is used to represent each document in the collection. Term weighting is done after the construction of the document graph to estimate the significance of a term to the document [9]. Alhaj et al. presented a model for ATC using a three-classification algorithm to classify AT, which is affected by two types of representation, BoW and TF–IDF, on CNNDS Arabic corpus. At the same time, they used CHI to remove unnecessary features [173]. Abdelaal et al. proposed a system for categorizing hadith into different classes based on content. The best three classifiers assessed primarily are DT with 0.965%, RF with 0.956%, and NB with 0.951% [176]. Daher et al. aimed to introduce a simple approach for handling SA by extracting opinions from Arabic tweets using ML [185]. Abdelgwad et al. suggested DL based on GRU and an interactive BiLSTM network for classification [186].

All literature review comprehensively covers an extensive range of studies on ATC and ATR. By categorizing these studies into key themes such as feature extraction methods, classification techniques, and application areas, the review provides a structured understanding of the field. Furthermore, it includes an analysis of recent trends, such as the shift from traditional ML models to DL [187] architectures, and explores underrepresented challenges like dialectical Arabic processing.

8. Datasets

ATC models have used and utilized various datasets, only some of these datasets are available for public, and most of these datasets are not available. In addition, one of the problems for ATC is the lack of a benchmark dataset with a large size. In this section, we list datasets published for ATC by analyzing them based on the number of documents, number of class, number of words, and references of all datasets to help researchers as illustrated in Table 6.

The field of ATC relies on a variety of datasets, each with unique features and limitations. For instance, the AraSenTi dataset is widely used for SA, containing tweets labeled for polarity. However, it is limited in linguistic diversity, focusing primarily on MSA. Similarly, OSACT datasets emphasize dialectical Arabic but often over represent Egyptian and Levantine dialects, introducing bias in model training. A critical evaluation of these datasets reveals common challenges, including unbalanced class distributions and the prevalence of informal text, such as social media posts with spelling errors and code-switching. These issues highlight the need for more comprehensive and diverse datasets to advance the field of ATC.

9. Tools and Open-Source Library

There are different tools and open sources available for ATC models. In addition, one of the problems for ATC is the lack of open sources. In this section, we list some of these resources for ATC by analyzing them based on the name, and references of all resources to support researchers, as we illustrate in Table 7.

Table 6: Summary of dataset and corpus available.

Ref.	Year	Dataset name	Class	Word	Document	Remark	Utilization	Website
[188]	2010	CNN	6	2,241,348	5070	OSAC	14	https://sourceforge.net/projects/ar-text-mining/files/Arabic-Corpora/
[188]	2010	BBC	7	1,860,786	4763	OSAC	8	https://sourceforge.net/projects/ar-text-mining/files/Arabic-Corpora/
[188]	2010	OSAc	10	18,183,511	22,429	OSAC	15	https://sourceforge.net/projects/ar-text-mining/files/Arabic-Corpora/
[189]	2014	LABR	2&5	63,000	8,520,886	Sentiment analysis/ classification	6	https://github.com/mohamedadaly/LABR
[190]	2019	Alkhaleej	7	—	45,500	SANAD	2	https://data.mendeley.com/datasets/57zpx667y9/2
[190]	2019	NADIA1	24	—	678,563	NADIA (multi label)	2	https://data.mendeley.com/datasets/hhrb7phdyx/2
[190]	2019	NADIA2	28	—	678,563	NADIA (multi label)	1	https://data.mendeley.com/datasets/hhrb7phdyx/2
[190]	2019	AKHBARONA	7	—	78,050	SANAD	2	https://data.mendeley.com/datasets/57zpx667y9/2
[190]	2019	ALARABIYA	6	—	71,247	SANAD	2	https://data.mendeley.com/datasets/57zpx667y9/2
[191]	2016	Abu El-Khair corpus	—	1,525,722,252	5,222,973	Corpus	NA	https://www.abuelkhair.net/index.php/en/arabic/abu-el-khair-corpus
[192]	2017	Tashkeela	—	75,629,921		Corpus	1	https://tashkeela.source forge.net
[8]	2018	M BINIZY	5	319,254,124	111,728	Document	1	https://data.mendeley.com/datasets/v524p5dhpj/2
[193]	2018	AL-HAJ	6	—	1000	Document	1	https://github.com/yalhag1/Alj-News-Arabic-text-classification-dataset
[194]		MANY	—	—	—	BY TAMER	—	https://qufaculty.qu.edu.qa/telsayed/datasets/
[195]	2020	BRAD-Arabic	2 & 3	39,886,898	510,598	Sentiment analysis/ classification	2	https://github.com/elnagara/BRAD-Arabic-Dataset
[196]	2020	HARD-Arabic	2&3	8,520,886	373,750	Sentiment analysis/ classification	2	https://github.com/elnagara/HARD-Arabic-Dataset
[197]	2015	TALAA	8	14,068,407	57,827	Document	—	https://github.com/saidziani/Arabic-News-Article-Classification
[198]	2022	Masader	—	—	—	Document	—	https://arbml.github.io/masader/
[1]	2018	Arabic corpus	—	1.9 B words	—	Corpus	—	https://archive.org/details/arabic_corpus
[2]	2020	arTenTen	—	10 B words	—	Corpus	—	https://www.sketchengine.eu/artenten-arabic-corpus/
[3]		GDELT project	—	9.5 B	—	Corpus	—	https://www.gdeltproject.org/

[1]https://archive.org/details/arabic_corpus.
[2]https://www.sketchengine.eu/artenten-arabic-corpus/.
[3]https://www.gdeltproject.org/.

Table 7: Summary of dataset and corpus available.

Description	Website
The "Rand" library has been launched to generate random ATs	https://tahadz.wordpress.com/2020/08/10/arrand/
A specific Arabic language library for Python provides basic functions to manipulate Arabic letters and text	https://pypi.org/project/PyArabic/
Fine-tuning BERT models for Arabic dialect detection	https://github.com/issam9/finetuning-bert-models-for-arabic-dialect-detection
At QCRI, we are dedicated to promoting the Arabic language in the information age by conducting world-class research in Arabic language technologies	https://alt.qcri.org/
Building open-source NLP libraries and tools for the Arabic language	https://omdena.com/projects/nlp-arabic/
Arabic language support for Text Blob	https://github.com/adhaamehab/textblob-ar
It can used as library "see section Arabic stop words library"	https://pypi.org/project/Arabic-Stopwords/
Search Gumar for millions of words from Gulf Arabic	https://camel.abudhabi.nyu.edu/gumar/
IWAN strives to publish research that serves the society and contributes in building a knowledge economy, through establishing a motivating environment, perfect placement of technology, and effective local and international partnership	https://iwan.ksu.edu.sa/ar
Arabic NLP Survey Papers Repository (ASPR)—مستودع الأوراق المسحية في معالجة اللغة العربية (أسبر)	https://github.com/iwan-rg/ArabicSurvey
The goal of this project is to create an Arabic benchmark for multitask learning, similar to the GLUE benchmark	https://www.alue.org/home
ARABIC NLP TOOLS CATALOGUE is a catalog has 64 tools added by 8 contributors	https://arbml.github.io/adawat/

10. Quantitative Analysis

The tables and figures included in this review significantly enhance comprehension by summarizing complex information concisely, so in this section, we explore quantitative analyses of ATC and ATR. To kick off the survey process, we first formulated key research questions focused on the effectiveness of various ATC methods and the challenges specific to AT. The foundation of the survey was built on peer-reviewed studies from the last 5 years, ensuring that the most recent advancements in the field were considered. We used a systematic review methodology, selecting studies based on their relevance to AT processing and evaluating them through a comparative analysis of the methodologies and datasets used. This procedure provided a structured approach to understanding the current state of ATC. It has been seen that the number of publications is 179 articles and we do our study based on the main subcategory for each stage and publication year. Moreover, these analyses will answer the following questions:

- How many research articles were published in each subcategory (methods in each stage)?
- How many research articles were published in the timeline for 2002–2021?
- Which stage of the ATC models are studied most and the least?
- What does a distribution of the papers look like for each subcategory based on using methods?
- What does a distribution of the papers look like for each subcategory based on timeline?
- What are the available datasets for ATC?
- What are the advantages and disadvantages of ATC and ATR?
- What challenges and restrictions do ATC and ATR still have for the future?

There are 179 papers, papers in our taxonomy which are divided into four stages the percentage for each one is illustrated in Figure 7. The total number of surveyed articles, including survey papers in this taxonomy, is 179 articles, out of which 30.32% is related to representation, 29.68% is related to preprocessing, and 23.87% is related to classification, whereas the rest 16.13% is related to the DR, and we explore this clearly in Figure 7.

The examined research papers in this taxonomy were quantitatively assessed based on their main category stages to address the aforementioned research questions. Then, depending on their subcategories, each primary category was quantitatively examined. Finally, available datasets were qualitatively studied based on the number of documents, classes, words, and references. We observe that most categories that have been studied are representation and the less is DR.

10.1. Preprocessing. In this subsection, we quantitatively analyze the reviewed preprocessing techniques based on their categories and timeline as follows.

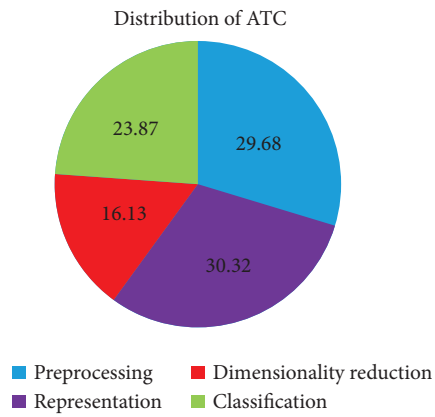

Figure 7: Distribution of ATC surveys and papers publications.

10.1.1. Preprocessing Techniques Based on Categories. The total number of reviewed research papers related to preprocessing is 46, which is 29.68% out of the total reviewed articles. However, Figure 8 shows the distribution of published papers among the preprocessing categories, tokenization, stop word, stemming, lemmatization, and hybrid, 50.55%, 2.2%, 39.56%, 2.2, and 5.94%, respectively. It can be observed that the tokenization category has obtained the highest percentage 50.55% because any processing for text has to tokenize text to a character or word, whereas lemmatization and stop words are 2.2% which is the lowest number of publications. Hybrid categories obtained have 5.49. Finally, stemming has a second value it can be concluded that the stemming category has been given more attention from all research after ignoring tokenization.

10.1.2. Preprocessing Techniques Based on Timeline. In this subsection, we quantitatively analyze the considered research papers in this article based on the timeline. Figure 9 shows the distribution of published papers on the timeline starting from 2001 to 2021. However, it can be seen that from the period 2001 to 2010, 28.26% of papers were published out of 45; it is distributed over 10 years, which is half of the considered period, which is why it has obtained the highest percentage. On the other side, 2015 and 2016 obtained 10.87, which is the highest percentage. Also, the years 2014, 2017, and 2019 have the same score of 8.7%, but it is lower, whereas the years 2012, 2013, 2014, 2018, and 2020 have obtained equal percentages of 4.35%, which is the lowest value. It can be concluded that more attention was given to representation in the years 2015 and 2016.

10.2. Representation. In this subsection, we quantitatively analyze the reviewed representation methods based on their categories and timeline.

10.2.1. Representation Techniques Based on Categories. The total number of reviewed research papers related to representation is 47, which is 30.32% out of the total

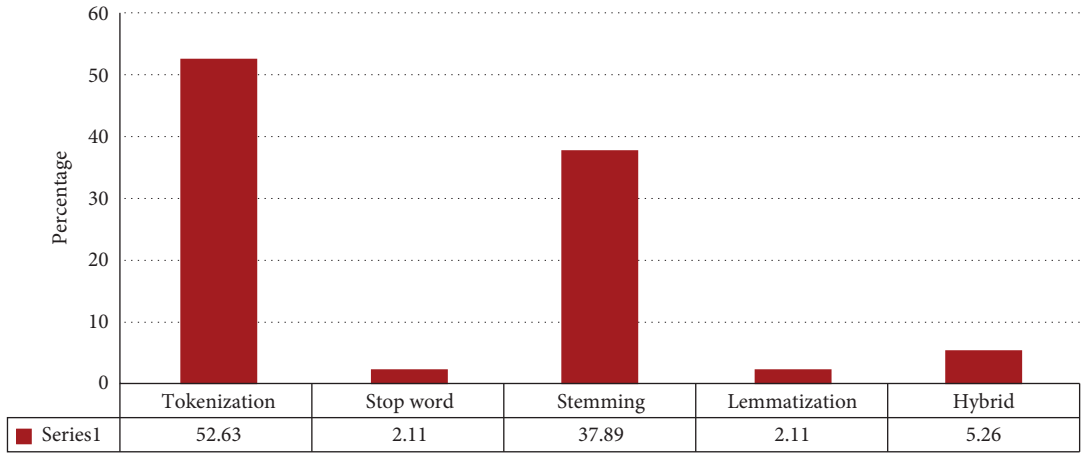

Figure 8: Using preprocessing techniques based on categories.

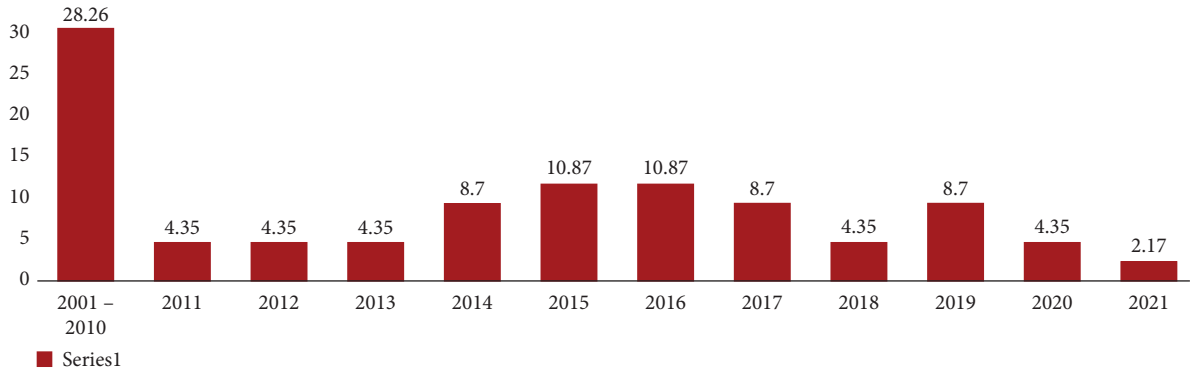

Figure 9: Using preprocessing techniques based on timeline.

reviewed articles. However, Figure 10 shows the distribution of published papers among the representation categories, char level, word, sentence, documents, and hybrid, 21.28%, 63.21%, 4.26%, 2.13%, and 8.51%, respectively. It can be observed that the word category has obtained the highest percentage, 63,83%, whereas the document level got 2.13%, which is the lowest number of publications. Sentence categories obtained the present age 4.26. To this end, it can be concluded that the word category has been given more attention in research than the other categories.

10.2.2. Representation Techniques Based on a Timeline. In this subsection, we quantitatively analyze the considered research papers in this article based on the timeline. Figure 11 shows the distribution of published papers on the timeline starting from 2001 to 2021. However, it can be seen that from the period 2001 to 2010, 4.26% of papers were published out of 47, which is distributed over 10 years, which is half of the considered period, and the highest percentage in 2020. On the other side, from 2011 to 2019, the numbers increased one after another, except that 2016 and 2017 were equal. Finally, it can be concluded that more attention was given to representation in the year 2020.

10.3. DR. In this subsection, we quantitatively analyze the reviewed DR methods based on their categories and timeline.

10.3.1. DR Techniques Based on Categories. The total number of reviewed research papers related to representation is 25, which is 16.13% of the total reviewed articles. However, Figure 12 shows the distribution of published papers among the representation categories, feature selection, feature extraction, and hybrid, 36%, 8%, 12%, and 44%, respectively. It can be observed that the hybrid category has obtained the highest percentage, 44%, whereas the feature extraction level got 8%, which is the lowest number of publications. Optimization categories obtained 12%. To this end, it can be concluded that the hybrid category has been given more attention in research than the other categories.

10.3.2. DR Techniques Based on the Timeline. In this subsection, we quantitatively analyze the considered research papers in this article based on the timeline. Figure 13 shows the distribution of published papers on the timeline starting from 2001 to 2021. However, it can be seen that from the period 2001 to 2010, 16% of papers were published out of 25, which is distributed between 10 years, which is half of the

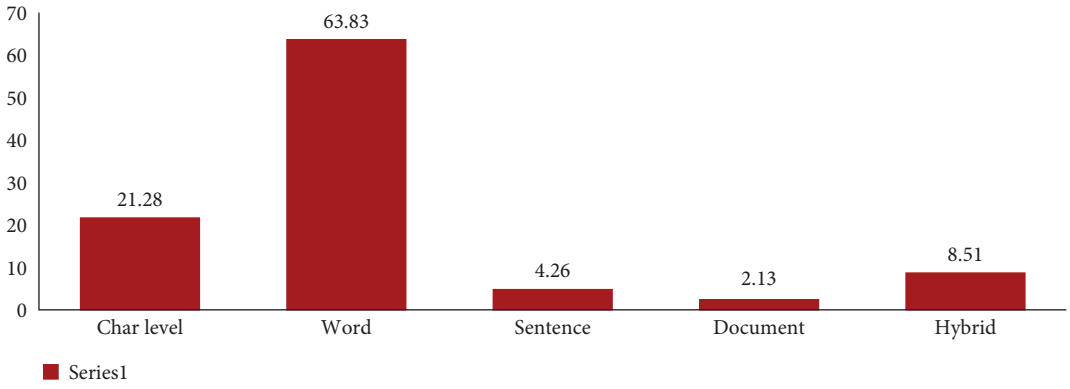

Figure 10: Using representation techniques based on categories.

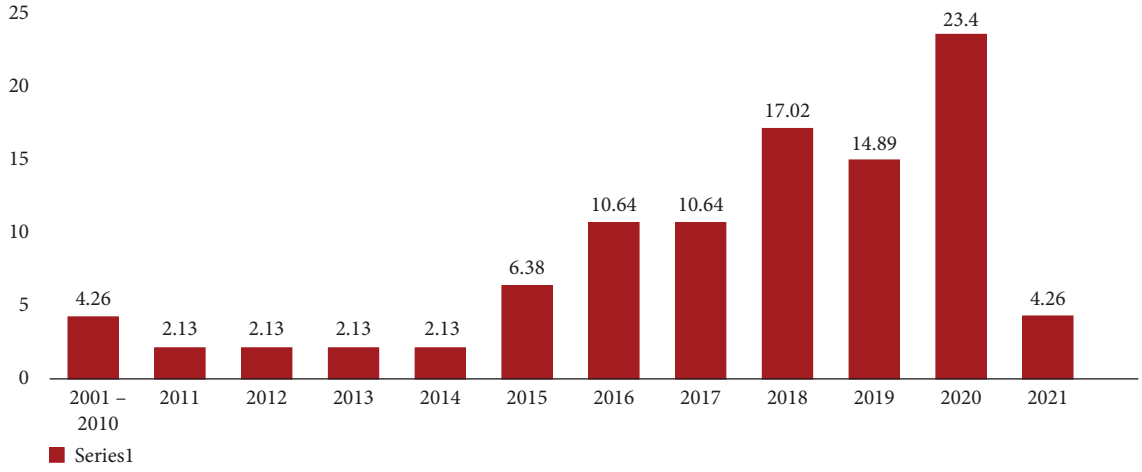

Figure 11: Using representation techniques based on timeline.

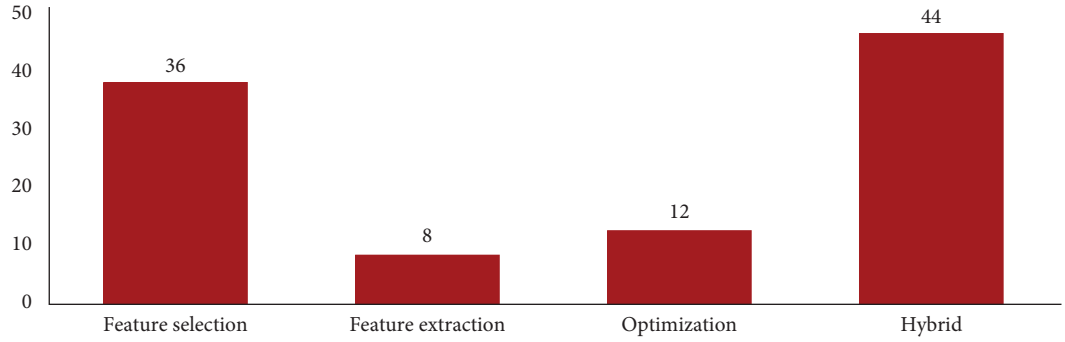

Figure 12: Using dimensionality reduction techniques based on categories.

considered period, that is, why it has obtained the next highest percentage after 2019. On the other side, year 2019 has obtained 20%, which is the highest percentage. Also, the years 2011, 2013, 2015, 2016, 2017, and 2021 have got the same score of 4%, but it is lesser; at the same time, 2020 and 2012 are the same percentage, and 2014 and 2018 are also the same percentages. It can be concluded that more attention was given to representation in the years 2019 followed by 2018 by neglecting the first value which considers 10 years from 2001 to 2010.

10.4. Classification. In this subsection, we quantitatively analyze the reviewed classification methods based on their categories and timeline.

10.4.1. Classification Techniques Based on Categories. The total number of reviewed research papers related to representation is 37 which is 32.87% out of the total reviewed articles. However, Figure 14 shows the distribution of published papers among the classification categories rule-

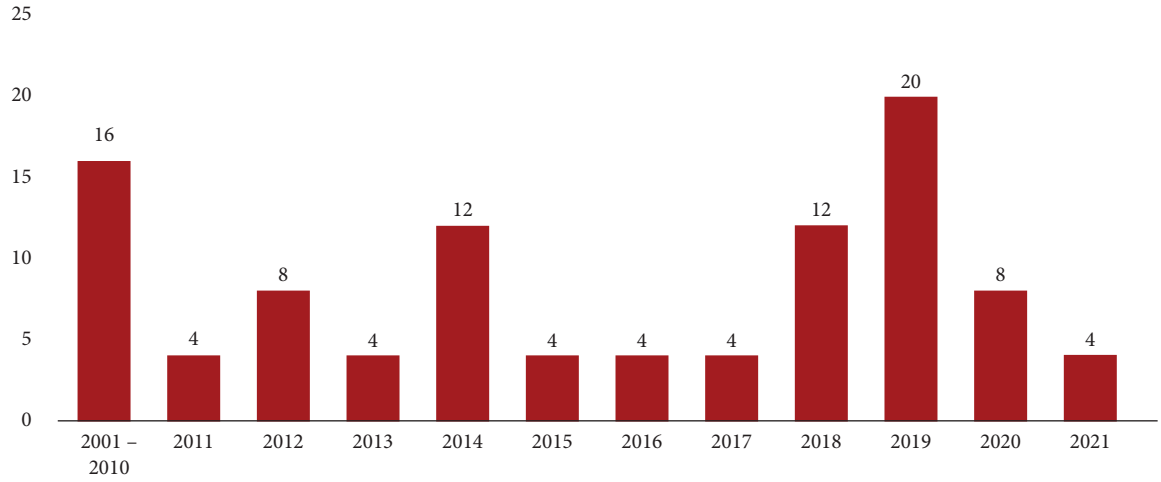

Figure 13: Using dimensionality reduction techniques based on timeline.

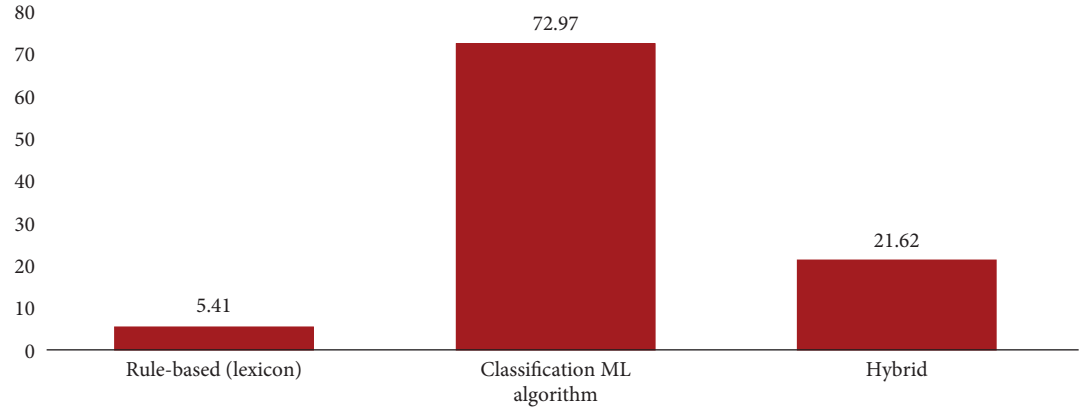

Figure 14: Classification techniques based on categories.

based lexicon, ML, and hybrid 5.41%, 72.97%, and 21.62%, respectively. It can be observed that the ML category has obtained the highest percentage 72.97%, whereas the rule-based level got 5.41%, which is the lowest number of publications. Hybrid categories obtained 21.62. To this end, it can be concluded that the ML category has been given more attention from research that the other categories, especially with DL at this time.

10.4.2. Classification Techniques Based on the Timeline. In this subsection, we quantitatively analyze the considered research papers in this article based on the timeline. Figure 15 shows the distribution of published papers on the timeline starting from 2001 to 2021. However, it can be seen that from the period 2001 to 2010, 18.92% of papers were published out of 37, and it is distributed between 10 years, which is half of the considered period, which is why it has obtained the highest percentage. On the other side, 2019 and 2020 obtained 16.22, which is the highest percentage. Also, the years 2011, 2011, and 2019 have got the same score of 2.7%, but it is lesser, whereas the years 2012, 2013, 2014, 2018, and 2021 have obtained different percentage values. It can be concluded that more attention was given to representation in the years 2019 and 2020.

11. Experimental Analysis

They have conducted analysis of various ATC and ATR methods with different ML algorithms, which were experimentally implemented, and the performance has been evaluated in terms of accuracy, precision, recall, and F-measure.

11.1. Metrics Evaluation. There are weighted objective metrics to evaluate the ATC system. We are going to mention them here recall, precision, accuracy, F1-score, Matthew's correlation coefficient (MCC), and negative predictive value (NPV) were used [16, 199]. The metrics are defined as follows:

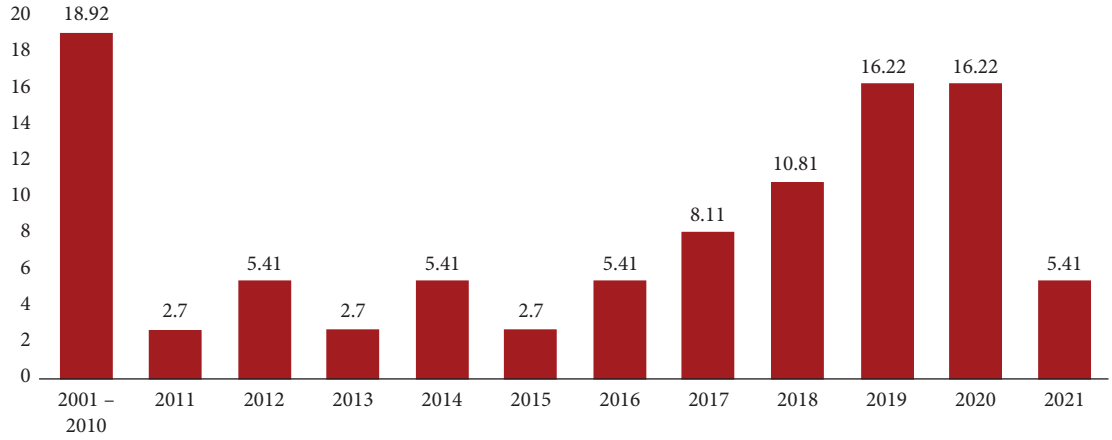

Figure 15: Classification techniques based on timeline.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{TN} + \text{FP}}, \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{3}$$

$$\text{F1} - \text{score} = \frac{2 \cdot \text{TP}}{2 \cdot \text{TP} + \text{FP} + \text{FN}}, \tag{4}$$

$$\text{MMC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}, \tag{5}$$

$$\frac{\text{Precision}}{\text{PPV}} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{6}$$

$$\text{NPV} = \frac{\text{TN}}{\text{TN} + \text{FN}}. \tag{7}$$

12. Discussion

This section describes the findings from the qualitative and quantitative analyses. Some results about existing ATC and ATR models were emphasized through qualitative investigation. Subsequently, general observations explore the merits and demerits of available models. Recent advancements in transformer-based models, such as ARAGPT, have shown great promise for Arabic text preprocessing. ARAGPT, designed specifically for Arabic, uses attention mechanisms to capture the complexities of the language, including its rich morphology and dialectal diversity. Compared to traditional methods, ARAGPT demonstrates superior performance in tasks like tokenization, normalization, and segmentation, all of which are crucial for ATC. Incorporating ARAGPT into the preprocessing pipeline can significantly enhance the accuracy and robustness of ATC systems. This survey explores these advanced techniques and compares them with other existing approaches in the field as we see similar work for other languages such as [200–203]. Finally, we discuss different observations as the following.

12.1. Qualitative and Qualitative Analysis. It is clear from Table 3 that there are many researchers have used ATR models. In addition, Table 4 investigates the existing ATC models and the actions to prepare for them. Table 3 explores the existing work for DR tasks which is less compared to representation and classification. However, prior research solved numerous issues, which we shall discuss in Section 10. On the other hand, it can be shown that DR has received less attention than preprocessing and classification. Quantitative analysis highlights some observations regarding the publications related to ATC based on the timeline, and stages of ATCs, where 2020 was the most productive year.

12.2. General Observations. Text classification involves many steps as we mentioned above and, in each stage, there are many algorithms have been used. In our study, we focused more on two steps, which affect the task of classification. These steps are representation and classification. We will mention some observations for these as follows in Table 8 and Table 9.

The field of ATC relies on a variety of datasets, each with unique features and limitations. For instance, the AraSenTi dataset is widely used for SA, containing tweets labeled for polarity. However, it is limited in linguistic diversity, focusing primarily on MSA. Similarly, OSACT datasets emphasize dialectical Arabic but often overrepresent Egyptian and Levantine dialects, introducing bias in model training. A critical evaluation of these datasets reveals common challenges, including unbalanced class distributions and the prevalence of informal text, such as social media posts with spelling errors and code-switching. These issues highlight the need for more comprehensive and diverse datasets to advance the field of ATC.

12.3. Open Issues and Challenges. Although the problem of automatic text classification enjoys quite a rich amount of literature, there are many challenges still open to research, including a focus on the lack of lexicons, lack of benchmark corpora, right-to-left reading, and compound phrases and idioms. There is a need for more efforts to implement modernized DL methods for ATC systems, while we have explored four AT steps (preprocessing, representation, DR, and classification) separately in Sections 4, 5, 6, and 7. Although there is some work has been done at ATC, the complexity of the Arabic language and lack of tools with an increasing number of documents make text processing and analysis a big data problem all of these prove that this topic still a hot area for a researcher. Furthermore, problems that are released to text in general such as representation feature extraction and selection still another option for research. We highlight in the following section a research gap that can facilitate a deeper understanding domain of ATC and improve these techniques. We list some of them in subsections as follows.

12.4. Challenges Related to Dataset, Lexicons, and Dictionaries

- The lack of publicly available free Arabic corpora.
- Lack of lexicons availability.
- Lack of dictionaries availability.
- Lack of data augmentation techniques for AT.

12.5. Challenges Related to Preprocessing

- Normalization process for the letter in some cases will change the meaning and affect accuracy for example "alif" (e.g., ا أ إ) is normalized to (ا) and will change the meaning; for example, فأر (means "mouse") will transform to فار (means "escaped")
- It is difficult to find roots of some words such as Arabized words, which are translated from other languages, for example, programs (برامج).
- In the Arabic language, one word may have more than one lexical category (noun, verb, adjective, etc.), for example, "eyes" ("الانسان عين") and wellspring (عين الماء), which makes it difficult to understand the meaning of AT.
- In the Arabic language, the problem of synonyms and broken plural forms is widespread which makes it difficult to recognize and understand the meaning of such words.
- Arabic letter Hamzah or Hamza (ء) can be written in four different forms (أ, ؤ, ئ, ء), so it is subjective to mistake and misuse with many words.
- Arabic nouns do not start with a capital letter as in English so another challenge for automatic AT processing which makes it difficult to recognize nouns Arabic language.
- Stemming problem of AT.

12.6. Challenges Related to Representation and Feature Engineering

- Curse of dimensionality and sparse vectors.
- Finding techniques that handle the context meaning of ATC.
- Time-consuming and memory space with new representation techniques such as BERT.

12.7. Challenges Related to Difficulties Nature of Arabic Language

- Orthographic, ambiguity, dialectal variation.
- Neglect many of Arabic language delicate such as Arabic, Khuzestan Arabic, Khurasan Arabic, Uzbekistan Arabic, the sub-Saharan Arabic of Nigeria and Chad, Djibouti Arabic, Cypriot Arabic, and Maltese.

12.8. Challenges Related to Related Topics

- Mixed language problem.
- Multimodal problem and multilanguage (mixed language) problem.
- For instance, Persian and Urdu both utilize the extended Arabic script, incorporating additional letters such as "ژ" "چ" "پ," and "گ" not found in standard Arabic. This often complicates applying Arabic-trained models to these languages without fine-tuning or transfer learning techniques.
- Kurdish, especially in its Sorani dialect, and Pashto also use modified Arabic scripts, and like Arabic, they suffer from issues like:
  i. Lack of diacritics in standard writing

Table 8: Observation of representation technique.

Strength/weakness	BOW	TF–IDF	W2V	GLOVE	GLOVET	FAST	CONTEXT
Easy to compute	✓	✓					
Compute the similarity			✓	✓	✓	✓	✓
Syntactic			✓	✓	✓	✓	✓
Semantics			✓	✓	✓	✓	✓
Capture polysemy						✓	✓
Capture out-of-vocabulary						✓	
Memory consumption			✓	✓	✓	✓	✓
Work on only sentence level							✓
Need on huge corpus to train					✓	✓	✓
Context handling							✓

Table 9: Observation of classification technique.

Strength/weakness	RA	BBA	LRA	NBA	KNN	SVM	DT	CRF	RF	DL
Easy to implement	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Robust		✓	✓	✓	✓	✓	✓	✓	✓	✓
Flexible with feature design	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Expensive to train										✓
Finding an efficient architecture is difficult										✓
Is it a fast algorithm	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Is it a black-box										✓
Handle online learning										✓
Parallel processing capability										✓
Requires a large amount of data										✓

ii. Highly inflected morphology
iii. Ambiguous word boundaries
iv. Scarcity of annotated corpora [204–206].

13. Conclusion

This study presented a comprehensive taxonomy review for ATC, which focuses on two main sections. In the beginning, a detailed analysis of the current ATC surveys based on their objective, functions, and methods has been done and compared with this study. Then, a survey for each topic was individually conducted such as preprocessing, representation, and classification. In addition, quantitative analysis has been done for each stage. Finally, the study briefly describes and lists the current open research challenges and future direction of the ATC system. There are many open challenges for ATC at every stage. In addition, future research directions are promising in this field such as multimodels, multilanguage models, and difficulties regarding the nature of the Arabic language such as dialectal, morphology, and stemming. So, based on our understanding, this study is helpful for the research community in finding gaps and challenges for the ATC system in the real scenario. It encourages researchers to develop an effective and efficient model for ATC in different domains such as healthcare, economics, business, and education. Ultimately, this study serves as a valuable resource for researchers by identifying key gaps and challenges in ATC and encouraging the development of more effective and efficient models. By addressing these challenges and exploring innovative approaches, future research can significantly enhance the capabilities of ATC systems, making them more robust and adaptable to real-world applications. In addition to the technical challenges and advancements in ATC, it is crucial to consider the ethical implications of applying ML to AT. One of the major difficulties in working with ATC is that there has been limited research addressing bias and ethical considerations in this field. Given these challenges, future research should focus on mitigating bias, ensuring dataset diversity, and developing explainable AI models to enhance fairness and accountability in ATC. Addressing these ethical considerations will be essential for building more trustworthy and responsible AI systems in this domain.

Nomenclature

ANN	Artificial neural networks
AT	Arabic text
ASA	Arabic sentiment analysis
ATC	Arabic text classification
ATR	Arabic text representation
CNN	Convolutional neural networks
DA	Dialect Arabic
DR	Dimensionality reduction
DT	Decision Tree
GRU	Gated recurrent units
IR	Information retrieval
K-NN	K-nearest neighbor
LDA	Latent Dirichlet allocation
LR	Logistic regression
LSTM	Long short-term memory
LSVC	Linear support vector classifier
ML	Machine learning
NB	Naive Bayes

NMF	Non-negative matrix factorization
OOV	Out-of-vocabulary
OSAC	Open-Source Arabic Corpus
PCA	Principal component analysis
RSG	Rich semantic graph
SA	Sentiment analysis
SVC	Support vector classifier
SVM	Support vector machines
TCW–ICF	Term class weight–inverse class frequency
TF	Term frequency
TF–IDF	Term frequency–inverse document frequency
UN	United Nations
VSM	Vector space model
BoW	Bag-of-words
TCR	Term class relevance
ISRI	Information Science Research Institute
SVD	Singular value decomposition
GWO	Grey wolf optimizer
ABC	Artificial bee colony
BPSO	Binary particle swarm optimization

Data Availability Statement

The data that support the findings of this study are available within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Author Contributions

Abdullah Y. Muaad: data curation, formal analysis, visualization, validation, software, and writing – original draft; Md Belal Bin Heyat and Faijan Akhtar: conceptualization, formal analysis, investigation, project administration, and writing – original draft; Usman Naseem and Wadeea R. Naji: data curation, validation, software, and writing – review and editing; Suresha Mallappa and Hanumanthappa J.: conceptualization, funding acquisition, supervision, investigation, and writing – review and editing. All authors read and agreed to the publication.

Funding

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

Acknowledgments

The authors would like to thank Prof. Sawan, Prof. Naseem, Prof. Lai, Prof. Singh, and Prof. Wu for their valuable help and support throughout this work.

References

[1] S. M. Grimes, M. A. Communication, and S. M. Grimes, *The Digital Child at Play: How Technological, Political and Commercial Rule Systems Shape Children's Play in Virtual Worlds* (School of Communication Examining Committee, 2010).

[2] A. Elnagar, S. M. Yagi, A. B. Nassif, I. Shahin, and S. A. Salloum, "Systematic Literature Review of Dialectal Arabic: Identification and Detection," *IEEE Access* 9 (2021): 31010–31042, https://doi.org/10.1109/ACCESS.2021.3059504.

[3] A. Y. Muaad, S. Raza, M. B. B. Heyat, A. Alabrah, and J. Hanumanthappa, "An Intelligent COVID-19-Related Arabic Text Detection Framework Based on Transfer Learning Using Context Representation," *International Journal of Intelligent Systems* 2024 (2024): 1–15, https://doi.org/10.1155/2024/8014111.

[4] K. Babić, S. Martinčić-Ipšić, and A. Meštrović, "Survey of Neural Text Representation Models," *Information* 11 (2020): 1–32, https://doi.org/10.3390/info11110511.

[5] H. Muhammad Zeeshan, A. Sultana, M. B. B. Heyat, et al., "A Machine Learning-Based Analysis for the Effectiveness of Online Teaching and Learning in Pakistan during COVID-19 Lockdown," *WORK: A Journal of Prevention, Assessment & Rehabilitation* 81 (2024): 1–19, https://doi.org/10.1177/10519815241308161.

[6] F. Kousar, A. Sultana, M. A. Albahar, et al., "A Cross-Sectional Study of Parental Perspectives on Children about COVID-19 and Classification Using Machine Learning Models," *Frontiers in Public Health* 12 (2024): 1373883, https://doi.org/10.3389/fpubh.2024.1373883.

[7] A. A. Altowayan and L. Tao, "Word Embeddings for Arabic Sentiment Analysis," in *2016 IEEE International Conference on Big Data (Big Data)* (Washington, DC, USA: IEEE, February 2016), 3820–3825, https://doi.org/10.1109/BigData.2016.7841054.

[8] S. Boukil, M. Biniz, F. E. Adnani, L. Cherrat, and A. E. E. Moutaouakkil, "Arabic Text Classification Using Deep Learning Technics," *International Journal of Grid and Distributed Computing* 11, no. 9 (2018): 103–114, https://doi.org/10.14257/ijgdc.2018.11.9.09.

[9] M. S. El Bazzi, D. Mammass, T. Zaki, and A. Ennaji, "A Graph Based Method for Arabic Document Indexing," *Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)* 2016 (2016): 308–312, https://doi.org/10.1109/SETIT.2016.7939885.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *International Conference on Learning Representations 2013* (Scottsdale, AZ: Track Proc, May 2013), 1–12.

[11] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Doha, Qatar, October 2014), 1532–1543, https://doi.org/10.3115/v1/d14-1162.

[12] O. Melamud, J. Goldberger, and I. Dagan, "context2vec: Learning Generic Context Embedding with Bidirectional LSTM, CoNLL 2016-20th SIGNLL Conf," in *Computer National Language Learning Proceedings* (Berlin, Germany, August 2016), 51–61, https://doi.org/10.18653/v1/k16-1006.

[13] G. Kanaan, R. Al-Shalabi, S. Ghwanmeh, and H. Al-Ma'adeed, "A Comparison of Text-Classification Techniques Applied to Arabic Text," *Journal of the American Society for Information Science and Technology* 60 (2009): 1836–1844, https://doi.org/10.1002/asi.20832.

[14] F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys* 34 (2002): 1–47, https://doi.org/10.1145/505282.505283.

[15] M. Suhail, *Representation and Classification of Text Data* (Univ. Mysore, 2019).

[16] Sumbul, A. Sultana, M. B. B. Heyat, et al., "Efficacy and Classification of Sesamum indicum Linn Seeds with Rosa Damascena Mill Oil in Uncomplicated Pelvic Inflammatory Disease Using Machine Learning," *Frontiers in Chemistry* 12 (2024): 1–21, https://doi.org/10.3389/fchem.2024.1361980.

[17] F. Akhtar, M. Belal Bin Heyat, A. Sultana, et al., "Medical Intelligence for Anxiety Research: Insights from Genetics, Hormones, Implant Science, and Smart Devices with Future Strategies," *WIREs Data Mining and Knowledge Discovery* 14, no. 6 (2024): e1552, https://doi.org/10.1002/widm.1552.

[18] M. J. A. Fazmiya, A. Sultana, M. B. B. Heyat, et al., "Efficacy of a Vaginal Suppository Formulation Prepared with Acacia Arabica (Lam.) Willd. Gum and Cinnamomum Camphora (L.) J. Presl. In Heavy Menstrual Bleeding Analyzed Using a Machine Learning Technique," *Frontiers in Pharmacology* 15 (2024): 1331622–1331623, https://doi.org/10.3389/fphar.2024.1331622.

[19] M. B. Bin Heyat, F. Akhtar, S. J. Abbas, et al., "Wearable Flexible Electronics Based Cardiac Electrode for Researcher Mental Stress Detection System Using Machine Learning Models on Single Lead Electrocardiogram Signal," *Biosensors* 12, no. 6 (2022): 427, https://doi.org/10.3390/bios12060427.

[20] M. B. Bin Heyat, D. Lai, K. Wu, et al., "Role of Oxidative Stress and Inflammation in Insomnia Sleep Disorder and Cardiovascular Diseases: Herbal Antioxidants and Anti-inflammatory Coupled with Insomnia Detection Using Machine Learning," *Current Pharmaceutical Design* 28, no. 45 (2022): 3618–3636, https://doi.org/10.2174/1381612829666221201161636.

[21] F. Akhtar, M. B. B. Heyat, S. Parveen, et al., "Early Coronary Heart Disease Deciphered via Support Vector Machines: Insights from Experiments," in *2023 20th International Computer. Conference on Wavelet Active. Media Technology and. Information Processing* (Chengdu, China: IEEE, December 2023), 1–7, https://doi.org/10.1109/ICCWAMTIP60502.2023.10387051.

[22] F. Harrag and E. El-Qawasmah, "Neural Network for Arabic Text Classification," in *The Second International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2009)* (London, UK: IEEE, August 2009), 778–783, https://doi.org/10.1109/ICADIWT.2009.5273841.

[23] R. Ayadi, M. Maraoui, and M. Zrigui, "A Survey of Arabic Text Representation and Classification Methods," *Research in Computing Science* 117, no. 1 (2016): 51–62, https://doi.org/10.13053/rcs-117-1-4.

[24] M. El-Masri, N. Altrabsheh, and H. Mansour, "Successes and Challenges of Arabic Sentiment Analysis Research: a Literature Review," *Social Network Analysis and Mining* 7 (2017): 54–22, https://doi.org/10.1007/s13278-017-0474-x.

[25] N. Boudad, R. Faizi, R. Oulad Haj Thami, and R. Chiheb, "Sentiment Analysis in Arabic: A Review of the Literature," *Ain Shams Engineering Journal* 9, no. 4 (2018): 2479–2490, https://doi.org/10.1016/j.asej.2017.04.007.

[26] A. M. F. Al Sbou, "A Survey of Arabic Text Classification Models," *International Journal of Informatics and Communication Technology* 8, no. 1 (2019): 25–4355, https://doi.org/10.11591/ijict.v8i1.pp25-28.

[27] M. Sayed, R. K. Salem, and A. E. Khder, "A Survey of Arabic Text Classification Approaches," *International Journal of Computer Applications in Technology* 59, no. 3 (2019): 236–251, https://doi.org/10.1504/IJCAT.2019.098601.

[28] M. E. M. Abo, R. G. Raj, and A. Qazi, "A Review on Arabic Sentiment Analysis: State-Of-The-Art, Taxonomy and Open Research Challenges," *IEEE Access* 7 (2019): 162008–162024, https://doi.org/10.1109/ACCESS.2019.2951530.

[29] G. Badaro, R. Baly, H. Hajj, et al., "A Survey of Opinion Mining in Arabic: A Comprehensive System Perspective Covering Challenges and Advances in Tools, Resources, Models, Applications, and Visualizations," *ACM Transactions on Asian and Low-Resource Language Information Processing* 18, no. 3 (2019): 1–52, https://doi.org/10.1145/3295662.

[30] A. H. Mohammad, "Arabic Text Classification: A Review," *Modern Applied Science* 13, no. 5 (2019): 88, https://doi.org/10.5539/mas.v13n5p88.

[31] M. A. Omari and M. A. Hajj, "Classifiers for Arabic NLP: Survey," *International Journal of Computational Complexity and Intelligent Algorithms* 1, no. 3 (2020): 231, https://doi.org/10.1504/ijccia.2020.105538.

[32] O. Oueslati, E. Cambria, M. B. HajHmida, and H. Ounelli, "A Review of Sentiment Analysis Research in Arabic Language," *Future Generation Computer Systems* 112 (2020): 408–430, https://doi.org/10.1016/j.future.2020.05.034.

[33] R. Ghaly, A. Elkorany, and C. A. Ezzat, "Hate Speech Detection in Arabic Text: Survey," *Procedia Computer Science* 244 (2024): 166–177, https://doi.org/10.1016/j.procs.2024.10.222.

[34] A. Y. Muaad, S. Raza, U. Naseem, and H. J. J. Davanagere, "Arabic Text Detection: a Survey of Recent Progress Challenges and Opportunities," *Applied Intelligence* 53, no. 24 (2023): 29845–29862, https://doi.org/10.1007/s10489-023-04992-9.

[35] Z. Jianqiang and G. Xiaolin, "Comparison Research on Text Pre-processing Methods on Twitter Sentiment Analysis," *IEEE Access* 5 (2017): 2870–2879, https://doi.org/10.1109/ACCESS.2017.2672677.

[36] Z. Alyafeai, M. S. Al-shaibani, M. Ghaleb, and I. Ahmad, "Evaluating Various Tokenizers for Arabic Text Classification," *Neural Processing Letters* 55, no. 3 (2023): 2911–2933, https://doi.org/10.1007/s11063-022-10990-8.

[37] M. B. Bin Heyat, D. Adhikari, F. Akhtar, et al., "Intelligent Internet of Medical Things for Depression: Current Advancements, Challenges, and Trends," *International Journal of Intelligent Systems* 2025, no. 1 (2025): https://doi.org/10.1155/int/6801530.

[38] A. Sultana, F. Akhtar, M. B. B. Heyat, et al., "Unveiling the Efficacy of Unani Medicine in Female Disorders through Machine Learning: Current Challenges and Opportunities," in *2023 20th International Computer. Conference on Wavelet Active. Media Technology and. Information Processing. (ICCWAMTIP 2023)* (Chengdu, China: IEEE, December 2023), 1–6, https://doi.org/10.1109/ICCWAMTIP60502.2023.10385245.

[39] L. Ren, Y. Liu, C. Ouyang, et al., "DyLas: A Dynamic Label Alignment Strategy for Large-Scale Multi-Label Text Classification," *Information Fusion* 120 (2025): 103081, https://doi.org/10.1016/j.inffus.2025.103081.

[40] T. Wang, B. Hou, J. Li, P. Shi, B. Zhang, and H. Snoussi, "TASTA: Text-Assisted Spatial and Temporal Attention Network for Video Question Answering," *Advanced Intelligent Systems* 5, no. 4 (2023): https://doi.org/10.1002/aisy.202200131.

[41] L. Jing, X. Fan, D. Feng, C. Lu, and S. Jiang, "A Patent Text-Based Product Conceptual Design Decision-Making Approach Considering the Fusion of Incomplete Evaluation Semantic and Scheme Beliefs," *Applied Soft Computing* 157 (2024): 111492, https://doi.org/10.1016/j.asoc.2024.111492.

[42] W. Song, Z. Ye, M. Sun, X. Hou, S. Li, and A. Hao, "AttriDiffuser: Adversarially Enhanced Diffusion Model for Text-To-Facial Attribute Image Synthesis," *Pattern*

Recognition 163 (2025): 111447, https://doi.org/10.1016/j.patcog.2025.111447.

[43] L. S. Larkey and M. E. Connel, *Automatic Information Retrieval at UMass in TREC-10, Tenth Text Retr* (Conf, 2001).

[44] L. S. Larkey, L. Ballesteros, and M. E. Connell, "Improving Stemming for Arabic Information Retrieval," in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (New York, NY: ACM Press, June 2002), 275–282, https://doi.org/10.1145/564376.564425.

[45] M. Aljlayl and O. Frieder, "On Arabic Search," in *CIKM' 02: Proceedings of the Eleventh International Conference on Information and Knowledge Management* (New York, NY: ACM, June 2002), 340–347, https://doi.org/10.1145/584792.584848.

[46] R. Duwairi, M. Al-Refai, and N. Khasawneh, "Stemming versus Light Stemming as Feature Selection Techniques for Arabic Text Categorization," in *International Conference on Innovations in Info-business & Technology (ICIIT 2023)* (Rome, Italy, August 2007), 446–450, https://doi.org/10.1109/IIT.2007.4430403.

[47] N. Mansour, R. A. Haraty, W. Daher, and M. Houri, "An Auto-Indexing Method for Arabic Text," *Information Processing & Management* 44, no. 4 (2008): 1538–1545, https://doi.org/10.1016/j.ipm.2007.12.007.

[48] E. Al-Shammari and J. Lin, "A Novel Arabic Lemmatization Algorithm," in *Proceedings of SIGIR 2008* (Singapore, July 2008), 113–118, https://doi.org/10.1145/1390749.1390767.

[49] E. T. Al-Shammari and J. Lin, "Towards an Error-free Arabic Stemming," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management* (New York, NY: ACM, May 2008), 9–15, https://doi.org/10.1145/1460027.1460030.

[50] Z. Kchaou and S. Kanoun, "Arabic Stemming with Two Dictionaries," in *2008 International Conference on Innovations in Information Technology* (Taipei, Taiwan, December 2008), 688–691, https://doi.org/10.1109/INNOVATIONS.2008.4781780.

[51] X. Liu, J. Zhang, and C. Guo, "Full-text Citation Analysis: A New Method to Enhance Scholarly Networks," *Journal of the American Society for Information Science and Technology* 64, no. 9 (2013): 1852–1863, https://doi.org/10.1002/asi.22883.

[52] Q. W. Bsoul and M. Mohd, "Effect of ISRI Stemming on Similarity Measure for Arabic Document Clustering," in *Lecture Notes in Computer Science (LNCS)* (2011), 584–593, https://doi.org/10.1007/978-3-642-25631-8_53.

[53] G. Kanaan, R. Al-Shalabi, M. Ababneh, and A. Al-Nobani, "Building an Effective Rule-Based Light Stemmer for Arabic Language to Inprove Search Effectiveness," in *2008 International Conference on Innovations in Information Technology* (Taipei, Taiwan, December 2008), 312–316, https://doi.org/10.1109/INNOVATIONS.2008.4781687.

[54] E. T. Al-Shammari, "Improving Arabic Document Categorization: Introducing Local Stem," in *International Conference on Intelligent Systems Design and Applications (ISDA)* (Cairo, Egypt, November 2010), 385–390, https://doi.org/10.1109/ISDA.2010.5687235.

[55] M. Saad, "The Impact of Text Preprocessing and Term Weighting on Arabic Text Classification," (2010), https://site.iugaza.edu.ps/msaad/files/2012/05/mksaad-Arabic-text-classification-MSc-Thesis-2010-rev9.pdf.

[56] B. Al-Shargabi, W. Al-Romimah, and F. Olayah, "A Comparative Study for Arabic Text Classification Algorithms Based on Stop Words Elimination," in *ACM International Conference Proceeding Series (ICPS)* (New York, NY, April 2011), 1–5, https://doi.org/10.1145/1980822.1980833.

[57] Y. Alhanini and M. J. A. Aziz, "The Enhancement of Arabic Stemming by Using Light Stemming and Dictionary-Based Stemming," *Journal of Software Engineering and Applications* 04, no. 09 (2011): 522–526, https://doi.org/10.4236/jsea.2011.49060.

[58] M. Hadni, A. Lachkar, and S. A. Ouatik, "A New and Efficient Stemming Technique for Arabic Text Categorization," *2012 International Conference on Multimedia Computing and Systems* 2012 (2012): 791–796, https://doi.org/10.1109/ICMCS.2012.6320308.

[59] T. El-Shishtawy and F. El-Ghannam, "An Accurate Arabic Root-Based Lemmatizer for Information Retrieval Purposes," (2012), https://arxiv.org/abs/1203.3584.

[60] S. M. Oraby, Y. El-Sonbaty, and M. A. El-Nasr, "Exploring the Effects of Word Roots for Arabic Sentiment Analysis," in *Proceedings of the Sixth International Joint Conference on Natural Language Processing* (Nagoya, Japan, October 2013), 471–479.

[61] M. N. Al-Kabi, "Towards Improving Khoja Rule-Based Arabic Stemmer," in *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)* (Amman, Jordan: IEEE, December 2013), 1–6, https://doi.org/10.1109/AEECT.2013.6716437.

[62] S. Bahassine, A. Madani, and M. Kissi, "Arabic Text Classification Using New Stemmer for Feature Selection and Decision Trees," *Journal of Engineering Science & Technology* 12 (2017): 1475–1487.

[63] M. N. Al-Kabi, S. A. Kazakzeh, B. M. Abu Ata, S. A. Al-Rababah, and I. M. Alsmadi, "A Novel Root Based Arabic Stemmer," *Journal of King Saud University-Computer and Information Sciences* 27, no. 2 (2015): 94–103, https://doi.org/10.1016/j.jksuci.2014.04.001.

[64] W. Salloum and N. Habash, "ADAM: Analyzer for Dialectal Arabic Morphology," *Journal of King Saud University-Computer and Information Sciences* 26, no. 4 (2014): 372–378, https://doi.org/10.1016/j.jksuci.2014.06.010.

[65] M. Pollak and M. Richard, "Suramin Blockade of Insulinlike Growth Factor I-Stimulated Proliferation of Human Osteosarcoma Cells," *JNCI Journal of the National Cancer Institute* 82, no. 16 (1990): 1349–1352, https://doi.org/10.1093/jnci/82.16.1349.

[66] S. A. Yousif, V. W. Samawi, I. Elkaban, and R. Zantout, "Enhancement of Arabic Text Classification Using Semantic Relations of Arabic WordNet," *Journal of Computer Science* 11 (2015): 498–509, https://doi.org/10.3844/jcssp.2015.498.509.

[67] T. Kanan and E. A. Fox, "Automated Arabic Text Classification with P-Stemmer, Machine Learning, and a Tailored News Article Taxonomy," *Journal of the Association for Information Science and Technology* 67, no. 11 (2016): 2667–2683, https://doi.org/10.1002/asi.23609.

[68] A. Nehar, D. Ziadi, and H. Cherroun, "Rational Kernels for Arabic Root Extraction and Text Classification," *Journal of King Saud University-Computer and Information Sciences* 28, no. 2 (2016): 157–169, https://doi.org/10.1016/j.jksuci.2015.11.004.

[69] A. Nehar, D. Ziadi, H. Cherroun, and Y. Guellouma, "An Efficient Stemming for Arabic Text Classification," in *2012 International Conference on Innovations in Information Technology (IIT)* (Abu Dhabi, UAE: IEEE, May 2012), 328–332, https://doi.org/10.1109/INNOVATIONS.2012.6207760.

[70] A. Al-Omari and B. Abuata, "Arabic Light Stemmer (ARS)," *Journal of Engineering Science & Technology* 9 (2014): 702–717.

[71] M. Hussein and M. Hussein, "Improving Arabic Text Categorization Using Normalization and Stemming Techniques," *International Journal of Computer Applications* 135, no. 2 (2016): 38–43, https://doi.org/10.5120/ijca2016908328.

[72] R. Mamoun and M. Ahmed, "Arabic Text Stemming: Comparative Analysis," in *2016 Conference of Basic Sciences and Engineering Studies* (Khartoum, Sudan, February 2016), 88–93, https://doi.org/10.1109/SGCAC.2016.7458011.

[73] A. Nasef and M. Jakovljević, "Development of Open-Source Software for Arabic Text Stemming and Classification," in *The 2016 International Science Conference* (Belgrade, Serbia: Singidunum University, May 2016), 271–276, https://doi.org/10.15308/sinteza-2016-271-276.

[74] A. Ayedh, G. Tan, K. Alwesabi, and H. Rajeh, "The Effect of Preprocessing on Arabic Document Categorization," *Algorithms* 9, no. 2 (2016): 27, https://doi.org/10.3390/a9020027.

[75] A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak, "Farasa: A Fast and Furious Segmenter for Arabic," in *The 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics* (San Diego, CA, June 2016), 11–16, https://doi.org/10.18653/v1/n16-3003.

[76] M. Mustafa, A. S. Eldeen, S. Bani-Ahmad, and A. O. Elfaki, "A Comparative Survey on Arabic Stemming: Approaches and Challenges," *Intelligent Information Management* 09, no. 02 (2017): 39–67, https://doi.org/10.4236/iim.2017.92003.

[77] K. Abainia, S. Ouamour, and H. Sayoud, "A Novel Robust Arabic Light Stemmer," *Journal of Experimental & Theoretical Artificial Intelligence* 29, no. 3 (2017): 557–573, https://doi.org/10.1080/0952813X.2016.1212100.

[78] M. T. B. Othman, M. A. Al-Hagery, and Y. M. E. Hashemi, "Arabic Text Processing Model: Verbs Roots and Conjugation Automation," *IEEE Access* 8 (2020): 103913–103923, https://doi.org/10.1109/ACCESS.2020.2999259.

[79] D. A. Said, N. M. Wanas, N. M. Darwish, and N. H. Hegazy, "A Study of Text Preprocessing Tools for Arabic Text Categorization," in *Second International Conference on Arabic Language Resources and Tools* (Tunis, Tunisia, April 2009), 230–236.

[80] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, "Feature Selection Using an Improved Chi-Square for Arabic Text Classification," *Journal of King Saud University-Computer and Information Sciences* 32, no. 2 (2020): 225–231, https://doi.org/10.1016/j.jksuci.2018.05.010.

[81] S. Boukil, F. El Adnani, A. E. El Moutaouakkil, L. Cherrat, and M. Ezziyyani, "Arabic Stemming Techniques as Feature Extraction Applied in Arabic Text Classification," *Lecture Notes in Networks and Systems* 25 (2018): 349–361, https://doi.org/10.1007/978-3-319-69137-4_31.

[82] Y. A. Alhaj, J. Xiang, D. Zhao, M. A. A. Al-Qaness, M. Abd Elaziz, and A. Dahou, "A Study of the Effects of Stemming Strategies on Arabic Document Classification," *IEEE Access* 7 (2019): 32664–32671, https://doi.org/10.1109/ACCESS.2019.2903331.

[83] A. S. Belal, "Comprehensive Processing for Arabic Texts to Extract Their Roots," *Iraqi Journal of Science* 60 (2019): 1404–1411, https://doi.org/10.24996/ijs.2019.60.6.25.

[84] O. Saoudi and R. Othman, "Retrieval Performance of Arabic Light Stemmers," *International Journal of Modern Trends in Social Sciences* 2, no. 10 (2019): 81–90, https://doi.org/10.35631/ijmtss.210008.

[85] D. Namly, "A Bi-technical Analysis for Arabic Stop-Words Detection," *Compusoft: An International Journal of Advanced Computer Technology* 8 (2019): 3126–3134.

[86] Y. A. Alhaj, M. A. A. Al-qaness, A. Dahou, M. Abd Elaziz, D. Zhao, and J. Xiang, "Effects of Light Stemming on Feature Extraction and Selection for Arabic Documents Classification," *Studies in Computational Intelligence* (2020): 59–79, https://doi.org/10.1007/978-3-030-34614-0_4.

[87] H. A. Almuzaini and A. M. Azmi, "Impact of Stemming and Word Embedding on Deep Learning-Based Arabic Text Categorization," *IEEE Access* 8 (2020): 127913–127928, https://doi.org/10.1109/ACCESS.2020.3009217.

[88] M. O. Hegazi, Y. Al-Dossari, A. Al-Yahy, A. Al-Sumari, and A. Hilal, "Preprocessing Arabic Text on Social Media," *Heliyon* 7, no. 2 (2021): e06191, https://doi.org/10.1016/j.heliyon.2021.e06191.

[89] M. Modhaffer and C. V Sivaramakrishna, "Prepositional Verbs in Arabic: A Corpus-Based Study," *Language India* 17 (2017): 154, https://www.researchgate.net/publication/320677565_Prepositional_Verbs_in_Arabic_A_Corpus-based_Study.

[90] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text Classification Algorithms: A Survey," *Information* 10 (2019): 1–68, https://doi.org/10.3390/info10040150.

[91] H. Bouamor, N. Habash, M. Salameh, et al., "The Madar Arabic Dialect Corpus and Lexicon," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (Miyazaki, Japan, May 2019), 3387–3396.

[92] F. Elghannam, "Text Representation and Classification Based on Bi-gram Alphabet," *Journal of King Saud University-Computer and Information Sciences* 33, no. 2 (2021): 235–242, https://doi.org/10.1016/j.jksuci.2019.01.005.

[93] L. Khreisat, "A Machine Learning Approach for Arabic Text Classification Using N-Gram Frequency Statistics," *Journal of Informetrics* 3, no. 1 (2009): 72–77, https://doi.org/10.1016/j.joi.2008.11.005.

[94] A. F. A. Nwesri, S. M. M. Tahaghoghi, and F. Scholer, "Capturing Out-Of-Vocabulary Words in Arabic Text," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing-EMNLP'06* (Sydney, Australia, October 2006), 258–266, https://doi.org/10.3115/1610075.1610113.

[95] N. Farra, E. Challita, R. A. Assi, and H. Hajj, "Sentence-level and Document-Level Sentiment Mining for Arabic Texts," in *2010 IEEE International Conference on Data Mining Workshops* (Abu Dhabi, UAE, December 2010), 1114–1119, https://doi.org/10.1109/ICDMW.2010.95.

[96] A. Karima, E. Zakaria, and T. G. Yamina, "Arabic Text Categorization: A Comparative Study of Different Representation Modes," *Journal of Theoretical and Applied Information Technology* 38 (2012): 1–5.

[97] S. S. Ismail, I. F. Moawad, and M. Aref, "Arabic Text Representation Using Rich Semantic Graph: A Case Study, Recent Adv. Inf. Sci" (2013).

[98] A. Alahmadi, A. Joorabchi, and A. E. Mahdi, "Arabic Text Classification Using Bag-Of-Concepts Representation," in *International Conference on Knowledge Discovery and Information Retrieval (KDIR 2014)* (Rome, Italy: SCITEPRESS-Science and and Technology Publications, October 2014), 374–380, https://doi.org/10.5220/0005138103740380.

[99] A. A. Al Sallab, H. Hajj, G. Badaro, R. Baly, W. El Hajj, and K. Bashir Shaban, "Deep Learning Models for Sentiment Analysis in Arabic," in *Proceedings of the Second Workshop on Arabic Natural Language Processing* (Beijing, China, August 2015), 9–17, https://doi.org/10.18653/v1/w15-3202.

[100] A. Al-Thubaity, M. Alhoshan, and I. Hazzaa, "Using Word N-Grams as Features in Arabic Text Classification," *Studies in Computational Intelligence* 569 (2015): 35–43, https://doi.org/10.1007/978-3-319-10389-1_3.

[101] A. Gelbukh, "Computational Linguistics and Intelligent Text Processing: 16th International Conference, CICLing 2015 Cairo, Egypt, April 14–20, 2015 Proceedings, Part I, Lect," *Notes Computer Science* 9041 (2015): 430–443, https://doi.org/10.1007/978-3-319-18111-0.

[102] F. S. Al-Anzi and D. AbuZeina, "Toward an Enhanced Arabic Text Classification Using Cosine Similarity and Latent Semantic Indexing," *Journal of King Saud University-Computer and Information Sciences* 29, no. 2 (2017): 189–195, https://doi.org/10.1016/j.jksuci.2016.04.001.

[103] A. Dahou, S. Xiong, J. Zhou, M. H. Haddoud, and P. Duan, "Word Embeddings and Convolutional Neural Network for Arabic Sentiment Classification," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (Osaka, Japan, December 2016), 2418–2427.

[104] Y. Belinkov and J. Glass, "A Character-Level Convolutional Neural Network for Distinguishing Similar Languages and Dialects," (2016), https://arxiv.org/abs/1609.07568.

[105] M. Hadni and M. Gouiouez, "Graph Based Representation for Arabic Text Categorization," in *ACM International Conference Proceeding Series (ICPS)* (New York, NY: ACM, June 2017), 1–7, https://doi.org/10.1145/3090354.3090431.

[106] A. El Mahdaouy, E. Gaussier, and S. O. El Alaoui, "Arabic Text Classification Based on Word and Document Embeddings," in *Advanced Intelligent Systems* (New York, NY: Springer International Publishing, 2017), 32–41, https://doi.org/10.1007/978-3-319-48308-5_4.

[107] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, "AraVec: A Set of Arabic Word Embedding Models for Use in Arabic NLP," *Procedia Computer Science* 117 (2017): 256–265, https://doi.org/10.1016/j.procs.2017.10.117.

[108] M. Gridach, H. Haddad, and H. Mulki, "Empirical Evaluation of Word Representations on Arabic Sentiment Analysis," *Communications in Computer and Information Science* 782 (2018): 147–158, https://doi.org/10.1007/978-3-319-73500-9_11.

[109] S. Al-Azani and E. S. M. El-Alfy, "Using Word Embedding and Ensemble Learning for Highly Imbalanced Data Sentiment Analysis in Short Arabic Text," *Procedia Computer Science* 109 (2017): 359–366, https://doi.org/10.1016/j.procs.2017.05.365.

[110] S. Mahmood and F. M. L. Al-Rufaye, "Arabic Text Mining Based on Clustering and Coreference Resolution," in *2017 International Conference on Current Research in Computer Science and Information Technology (ICCIT)* (Dhaka, Bangladesh, December 2017), 140–144, https://doi.org/10.1109/CRCSIT.2017.7965549.

[111] D. Sagheer and F. Sukkar, "Arabic Sentences Classification via Deep Learning," *International Journal of Computer Applications* 182, no. 5 (2018): 40–46, https://doi.org/10.5120/ijca2018917555.

[112] M. S. El Bazzi, D. Mammass, T. Zaki, and A. Ennaji, "Graph-based Text Modeling: Considering Mathematical Semantic Linking to Improve the Indexation of Arabic Documents," https://link.springer.com/series/0558?srsltid=AfmBOorxqn7EqVUFA67VdfvwS-nZDYtm9rOQJJTpm6GgqBu_Y31DYMqm.

[113] M. Ali, "Character Level Convolutional Neural Network for German Dialect Identification," in *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)* (Santa Fe, NM, August 2018), 172–177, https://aclanthology.org/W18-3913.

[114] D. S. Guru, M. Ali, and M. Suhil, "A Novel Term Weighting Scheme and an Approach for Classification of Agricultural Arabic Text Complaints," in *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)* (Manila, Philippines, April 2018), 24–28, https://doi.org/10.1109/ASAR.2018.8480317.

[115] D. Suleiman and A. Awajan, "Comparative Study of Word Embeddings Models and Their Usage in Arabic Language Applications," in *2018 International Arab Conference on Information Technology (ACIT)* (Werdanye, Lebanon, November 2018), 1–7, https://doi.org/10.1109/ACIT.2018.8672674.

[116] A. Alwehaibi and K. Roy, "Comparison of Pre-trained Word Vectors for Arabic Text Classification Using Deep Learning Approach," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)* (Orlando, FL, December 2018), 1471–1474, https://doi.org/10.1109/ICMLA.2018.00239.

[117] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "Improving Sentiment Analysis in Arabic Using Word Representation," in *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)* (Manila, Philippines, April 2018), 13–18, https://doi.org/10.1109/ASAR.2018.8480191.

[118] Q. A. Al-Radaideh and M. A. Al-Abrat, "An Arabic Text Categorization Approach Using Term Weighting and Multiple Reducts," *Soft Computing* 23, no. 14 (2019): 5849–5863, https://doi.org/10.1007/s00500-018-3249-z.

[119] M. M. Fouad, A. Mahany, N. Aljohani, R. A. Abbasi, and S. U. Hassan, "ArWordVec: Efficient Word Embedding Models for Arabic Tweets," *Soft Computing* 24, no. 11 (2020): 8061–8068, https://doi.org/10.1007/s00500-019-04153-6.

[120] H. Mulki, H. Haddad, M. Gridach, and I. Babaoğlu, "Syntax-ignorant N-Gram Embeddings for Dialectal Arabic Sentiment Analysis," *Natural Language Engineering* 27, no. 3 (2021): 315–338, https://doi.org/10.1017/S135132492000008X.

[121] E. Omara, M. Mosa, and N. Ismail, "Emotion Analysis in Arabic Language Applying Transfer Learning," in *2019 15th International Computer Engineering Conference (ICENCO)* (Cairo, Egypt, December 2019), 204–209, https://doi.org/10.1109/ICENCO48310.2019.9027295.

[122] F. Z. El-Alami and S. O. El Alaoui, "Word Sense Representation Based-Method for Arabic Text Categorization," in *The 9th International Symposium on Signal, Image, Video and Communications ISIVC 2018* (Rabat, Morocco, November 2018), 141–146, https://doi.org/10.1109/ISIVC.2018.8709234.

[123] R. M. K. Saeed, S. Rady, and T. F. Gharib, "An Ensemble Approach for Spam Detection in Arabic Opinion Texts," *Journal of King Saud University-Computer and Information Sciences* 34, no. 1 (2022): 1407–1416, https://doi.org/10.1016/j.jksuci.2019.10.002.

[124] W. Etaiwi and A. Awajan, "Graph-based Arabic Text Semantic Representation," *Information Processing & Management* 57, no. 3 (2020): 102183, https://doi.org/10.1016/j.ipm.2019.102183.

[125] A. T. Al-Taani and S. H. Al-Sayadi, "Classification of Arabic Text Using Singular Value Decomposition and Fuzzy C-Means Algorithms," *Algorithms for Intelligent Systems* (2020): 111–123, https://doi.org/10.1007/978-981-15-3357-0_8.

[126] G. M. B. Do Valle, "Engineering and Architecture," *Structural Engineering International* 4, no. 3 (1994): 141, https://doi.org/10.2749/101686694780601962.

[127] W. Antoun, F. Baly, and H. Hajj, "AraBERT: Transformer-Based Model for Arabic Language Understanding," (2020), https://arxiv.org/abs/2003.00104.

[128] A. I. Alharbi and M. Lee, *Combining Character and Word Embeddings for Affect in Arabic Informal Social Media Microblogs* (Springer International Publishing, 2020).

[129] S. A. Chowdhury, A. Abdelali, K. Darwish, J. Soon-Gyo, J. Salminen, and B. J. Jansen, "Improving Arabic Text Categorization Using Transformer Training Diversification," *Proceedings of the Fifth Arabic Natural Language Processing Workshop* 9 (2020): 226–236, https://aclanthology.org/2020.wanlp-1.21.

[130] Y. Li, Q. Pan, T. Yang, S. Wang, J. Tang, and E. Cambria, "Learning Word Representations for Sentiment Analysis," *Cognitive Computation* 9, no. 6 (2017): 843–851, https://doi.org/10.1007/s12559-017-9492-2.

[131] H. Elzayady, K. M. Badran, and G. I. Salama, "Arabic Opinion Mining Using Combined CNN-LSTM Models," *International Journal of Intelligent Systems and Applications* 12, no. 4 (2020): 25–36, https://doi.org/10.5815/ijisa.2020.04.03.

[132] V. Ramesh, V. C. Jaunky, R. Roopchund, and H. S. Oodit, "Customer Satisfaction, Loyalty and 'Adoption' of E-Banking Technology in Mauritius," *Advances in Intelligent Systems and Computing* (2021): 885–897, https://doi.org/10.1007/978-981-15-5400-1_84.

[133] A. K. Sharma, S. Chaurasia, and D. K. Srivastava, "Sentimental Short Sentences Classification by Using CNN Deep Learning Model with Fine Tuned Word2Vec," *Procedia Computer Science* 167 (2020): 1139–1147, https://doi.org/10.1016/j.procs.2020.03.416.

[134] H. Elfaik and E. H. Nfaoui, "Deep Bidirectional LSTM Network Learning-Based Sentiment Analysis for Arabic Text," *Journal of Intelligent Systems* 30, no. 1 (2020): 395–412, https://doi.org/10.1515/jisys-2020-0021.

[135] F. z. El-Alami, S. Ouatik El Alaoui, and N. En Nahnahi, "Contextual Semantic Embeddings Based on Fine-Tuned AraBERT Model for Arabic Text Multi-Class Categorization," *Journal of King Saud University-Computer and Information Sciences* 34, no. 10 (2022): 8422–8428, https://doi.org/10.1016/j.jksuci.2021.02.005.

[136] M. A. El-Affendi, K. Alrajhi, and A. Hussain, "A Novel Deep Learning-Based Multilevel Parallel Attention Neural (MPAN) Model for Multidomain Arabic Sentiment Analysis," *IEEE Access* 9 (2021): 7508–7518, https://doi.org/10.1109/ACCESS.2021.3049626.

[137] F. S. Al-Anzi and S. T. B. Shalini, "Revealing the Next Word and Character in Arabic: An Effective Blend of Long Short-Term Memory Networks and ARABERT," *Applied Sciences* 14, no. 22 (2024): 10498, https://doi.org/10.3390/app142210498.

[138] G. Petasis, F. Vichot, F. Wolinski, G. Paliouras, V. Karkaletsis, and C. D. Spyropoulos, "Using Machine Learning to Maintain Rule-Based Named-Entity Recognition and Classification Systems," in *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics ACL'01* (Morristown, NJ, May 2001), 426–433, https://doi.org/10.3115/1073012.1073067.

[139] M. Abdul-Mageed, A. R. Elmadany, and E. M. B. Nagoudi, "ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (May 2021), 7088–7105, https://doi.org/10.18653/v1/2021.acl-long.551.

[140] S. Sonawane, P. Kulkarni, and P. A. Kulkarni, "Graph Based Representation and Analysis of Text Document: A Survey of Techniques," *International Journal of Computer Applications* 96, no. 19 (2014): 1–8, https://doi.org/10.5120/16899-6972.

[141] M. Masih and A. Grant, "Chi Square Feature Extraction Based SVMS Arabic Language Text Categorization System," *TalentExcel: Talent Discovery Platform* 9 (2017): 18–26, https://doi.org/10.3844/jcssp.2007.430.435.

[142] A. M. d. Mesleh, "Support Vector Machines Based Arabic Language Text Classification System: Feature Selection Comparative Study," *Advances in Computer and Information Sciences and Engineering* (2008): 11–16, https://doi.org/10.1007/978-1-4020-8741-7_3.

[143] F. Thabtah, M. A. H. Eljinini, M. Zamzeer, and W. M. Hadi, "Naïve Bayesian Based on Chi Square to Categorize Arabic Data," *Innovation Knowledge Management* 1–3 (2009): 930–935.

[144] A. Alajmi, E. Saad, M. Awadalla, E. Saad, and M. Awadalla, "DACS Dewey Index-Based Arabic Document Categorization System," *International Journal of Computer Applications* 47, no. 23 (2012): 50–57, https://doi.org/10.5120/7500-0634.

[145] M. Zrigui, R. Ayadi, M. Mars, and M. Maraoui, "Arabic Text Classification Framework Based on Latent Dirichlet Allocation," *Journal of Computing and Information Technology* 20, no. 2 (2012): 125–140, https://doi.org/10.2498/cit.1001770.

[146] C. Sciences, "New Techniques for Arabic Document Classi Cation Hamouda Khalifa Hamouda Chantar" (2013).

[147] T. Zaki, Y. Es-saady, D. Mammass, A. Ennaji, and S. Nicolas, "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 (2014): 127–144, https://doi.org/10.14257/ijseia.2014.8.2.13.

[148] A. Abu-Errub, "Arabic Text Classification Algorithm Using TFIDF and Chi Square Measurements," *International Journal of Computer Applications* 93, no. 6 (2014): 40–45, https://doi.org/10.5120/16223-5674.

[149] A. H. Mohammad, "Applytwo Feature Selections (Chi-Square and Symmetric Uncertainty) Using C4. 5 Classification Algorithm Based on Arabic," *Technology* 2 (2019): 4–8.

[150] S. Larabi Marie-Sainte and N. Alalyani, "Firefly Algorithm Based Feature Selection for Arabic Text Classification," *Journal of King Saud University-Computer and Information Sciences* 32 (2020): 320–328, https://doi.org/10.1016/j.jksuci.2018.06.004.

[151] A. M. D. E. Hassanein and M. Nour, "A Proposed Model of Selecting Features for Classifying Arabic Text," *Jordanian Journal of Computers and Information Technology* 5, no. 0 (2019): 1–290, https://doi.org/10.5455/jjcit.71-1564059469.

[152] M. Belazzoug, M. Touahria, F. Nouioua, and M. Brahimi, "An Improved Sine Cosine Algorithm to Select Features for Text Categorization," *Journal of King Saud University-Computer and Information Sciences* 32, no. 4 (2020): 454–464, https://doi.org/10.1016/j.jksuci.2019.07.003.

[153] A. A. Mohamed, "An Effective Dimension Reduction Algorithm for Clustering Arabic Text," *Egyptian Informatics Journal* 21 (2020): 1–5, https://doi.org/10.1016/j.eij.2019.05.002.

[154] A. Adel, N. Omar, M. Albared, and A. Al-Shabi, "Feature Selection Method Based on Statistics of Compound Words for Arabic Text Classification," *The International Arab Journal of Information Technology* 16 (2019): 178–185.

[155] T. Sabbah and A. Selamat, "Intelligent Software Methodologies, Tools and Techniques," *Communications in Computer and Information Science* 532 (2015): 175–189, https://doi.org/10.1007/978-3-319-22689-7.

[156] M. Hijazi, A. Zeki, and A. Ismail, "Arabic Text Classification Using Hybrid Feature Selection Method Using Chi-Square Binary Artificial Bee Colony Algorithm," *International Journal of Mathematics and Computer Science* 16 (2021): 213–228.

[157] H. Chantar, M. Mafarja, H. Alsawalqah, A. A. Heidari, I. Aljarah, and H. Faris, "Feature Selection Using Binary Grey Wolf Optimizer with Elite-Based Crossover for Arabic Text Classification," *Neural Computing & Applications* 32, no. 16 (2020): 12201–12220, https://doi.org/10.1007/s00521-019-04368-6.

[158] E. M. Saad, M. H. Awadalla, and A. Alajmi, "Arabic Verb Pattern Extraction," in *10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010)* (Kuala Lumpur, Malaysia, May 2010), 642–645, https://doi.org/10.1109/ISSPA.2010.5605427.

[159] M. El Kourdi, A. Bensaid, and T. Rachidi, "Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm," in *Work Computer Approaches to Arabic Script-Based Language-Semit'04* (Morristown, NJ: Association for Computational Linguistics, 2004), 51, https://doi.org/10.3115/1621804.1621819.

[160] M. M. Syiam, Z. T. Fayed, and M. B. Habib, "An Intelligent System for {A}rabic Text Classification," *Int. Journal Intelligent Computing Information Sciences* 6 (2006): 1–19.

[161] S. Al-Harbi, A. Almuhareb, A. Al-Thubaity, M. S. Khorsheed, and A. Al-Rajeh, "Automatic Arabic Text Classification, Text," (2008), https://eprints.ecs.soton.ac.uk/22254/.

[162] F. Thabtah, "VSMs with K-Nearest Neighbour to Categorise Arabic Text Data," *World Congress on Engineering and Computer Science* 2173 (2008): 22–25.

[163] A. Mustafa El-Halees and A. El-Halees, "Arabic Opinion Mining Using Combined Classification Approach Opinion Mining View Project Educational Data Mining View Project Arabic Opinion Mining Using Combined Classification Approach," in *2024 25th International Arab Conference on Information Technology (ACIT)* (Azrqa, Jordan, June 2011), 264–271, https://www.researchgate.net/publication/228467530.

[164] F. Thabtah, O. Gharaibeh, and R. Al-Zubaidy, "Arabic Text Mining Using Rule Based Classification," *Journal of Information and Knowledge Management* 11, no. 01 (2012): 1250006–1250010, https://doi.org/10.1142/S0219649212500062.

[165] E. Alaa, "A Comparative Study on Arabic Text Classification, Egypt," *Computer Science Journal* 20, no. 2 (2008): https://www.researchgate.net/publication/220058961_A_Comparative_Study_on_Arabic_Text_Classification/file/e0b49524347d872516.pdf.

[166] A. Alahmadi, A. e. Mahdi, and A. Joorabchi, "Combining Bag-Of-Words and Bag-Of-Concepts Representations for Arabic Text Classification," in *25th IET Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communities Technologies (ISSC 2014/CIICT 2014)* (2014), 343–348, https://doi.org/10.1049/cp.2014.0711.

[167] R. Al-Shalabi, G. Kanaan, and M. H. Gharaibeh, "Arabic Text Categorization Using kNN Algorithm," in *4th International Conference on Computer Science and Information Technology (COMSCI 2025)* (Amman, Jordan, June 2006), 5–7.

[168] M. Gridach, "Character-Aware Neural Networks for Arabic Named Entity Recognition for Social Media," (2016), https://aclanthology.org/W16-3703.

[169] I. S. I. Abuhaiba and H. M. Dawoud, "Combining Different Approaches to Improve Arabic Text Documents Classification," *International Journal of Intelligent Systems and Applications* 9, no. 4 (2017): 39–52, https://doi.org/10.5815/ijisa.2017.04.05.

[170] M. Al-Smadi, O. Qawasmeh, M. Al-Ayyoub, Y. Jararweh, and B. Gupta, "Deep Recurrent Neural Network vs. Support Vector Machine for Aspect-Based Sentiment Analysis of Arabic Hotels' Reviews," *Journal of Computational Science* 27 (2018): 386–393, https://doi.org/10.1016/j.jocs.2017.11.006.

[171] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "A Combined CNN and LSTM Model for Arabic Sentiment Analysis," *Lecture Notes in Computer Science* (New York, NY, January 2018, 2018): 179–191, https://doi.org/10.1007/978-3-319-99740-7_12.

[172] M. Abdullah, M. Hadzikadicy, and S. Shaikhz, "SEDAT: Sentiment and Emotion Detection in Arabic Text Using CNN-LSTM Deep Learning," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)* (Orlando, FL, December 2018), 835–840, https://doi.org/10.1109/ICMLA.2018.00134.

[173] Y. A. Alhaj, W. U. Wickramaarachchi, A. Hussain, M. A. A. Alqaness, and H. M. Abdelaal, "Efficient Feature Representation Based on the Effect of Words Frequency for Arabic Documents Classification," in *Proceedings of the 2nd International Conference on Telecommunications and Communication Engineering* (Ottawa, Canada, April 2018), 397–401, https://doi.org/10.1145/3291842.3291900.

[174] K. Abu Kwaik, M. Saad, S. Chatzikyriakidis, and S. Dobnik, "LSTM-CNN Deep Learning Model for Sentiment Analysis of Dialectal Arabic," *Communications in Computer and Information Science* 1108 (2019): 108–121, https://doi.org/10.1007/978-3-030-32959-4_8.

[175] A. Elnagar, R. Al-Debsi, and O. Einea, "Arabic Text Classification Using Deep Learning Models," *Information Processing & Management* 57, no. 1 (2020): 102121, https://doi.org/10.1016/j.ipm.2019.102121.

[176] H. M. Abdelaal, B. R. Elemary, and H. A. Youness, "Classification of Hadith According to its Content Based on Supervised Learning Algorithms," *IEEE Access* 7 (2019): 152379–152387, https://doi.org/10.1109/ACCESS.2019.2948159.

[177] M. M. Al-Tahrawi and S. N. Al-Khatib, "Arabic Text Classification Using Polynomial Networks," *Journal of King Saud University-Computer and Information Sciences* 27, no. 4 (2015): 437–449, https://doi.org/10.1016/j.jksuci.2015.02.003.

[178] M. Alali, N. Mohd Sharef, M. A. Azmi Murad, H. Hamdan, and N. A. Husin, "Narrow Convolutional Neural Network for Arabic Dialects Polarity Classification," *IEEE Access* 7 (2019): 96272–96283, https://doi.org/10.1109/ACCESS.2019.2929208.

[179] M. Daif, S. Kitada, and H. Iyatomi, "AraDIC: Arabic Document Classification Using Image-Based Character Embeddings and Class-Balanced Loss," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop* (New York, NY, June 2020), 214–221, https://doi.org/10.18653/v1/2020.acl-srw.29.

[180] F. Z. El-Alami, A. El Mahdaouy, S. O. El Alaoui, and N. En-Nahnahi, "A Deep Autoencoder-Based Representation for Arabic Text Categorization," *Journal of Information and Communication Technology* 19 (2020): 381–398, https://doi.org/10.32890/jict2020.19.3.4.

[181] M. S. H. Ameur, R. Belkebir, and A. Guessoum, "Robust Arabic Text Categorization by Combining Convolutional and Recurrent Neural Networks," *ACM Transactions on*

Asian and Low-Resource Language Information Processing 19, no. 5 (2020): 1–16, https://doi.org/10.1145/3390092.

[182] F. Z. El-Alami, S. O. El Alaoui, and N. En-Nahnahi, "Deep Neural Models and Retrofitting for Arabic Text Categorization," *International Journal of Intelligent Information Technologies* 16, no. 2 (2020): 74–86, https://doi.org/10.4018/IJIIT.2020040104.

[183] M. Alhawarat and A. O. Aseeri, "A Superior Arabic Text Categorization Deep Model (SATCDM)," *IEEE Access* 8 (2020): 24653–24661, https://doi.org/10.1109/ACCESS.2020.2970504.

[184] A. H. Ombabi, W. Ouarda, and A. M. Alimi, "Deep Learning CNN–LSTM Framework for Arabic Sentiment Analysis Using Textual Information Shared in Social Networks," *Social Network Analysis and Mining* 10 (2020): 53–13, https://doi.org/10.1007/s13278-020-00668-1.

[185] S. Tareq Daher, A. Yunis Maghari, and H. Fares Abushawish, "Sentiment Analysis of Arabic Tweets on the Great March of Return Using Machine Learning," (2021), https://iugspace.iugaza.edu.ps/handle/20.500.12358/28675.

[186] M. Abdelgwad, T. H. Soliman, A. Taloba, M. F. Farghaly, A. I. Taloba, and M. F. Farghaly, "Arabic Aspect Based Sentiment Analysis Using Bidirectional GRU Based Models," *Journal of King Saud University-Computer and Information Sciences* 34, no. 9 (2022): 6652–6662, https://doi.org/10.1016/j.jksuci.2021.08.030.

[187] D. Li, K. D. Ortegas, and M. White, "Exploring the Computational Effects of Advanced Deep Neural Networks on Logical and Activity Learning for Enhanced Thinking Skills," *Systems* 11, no. 7 (2023): 319, https://doi.org/10.3390/systems11070319.

[188] M. Saad and W. Ashour, "OSAC: Open Source Arabic Corpora," in *6th International Conference on Electronics, Computer Engineering and Electrical Engineering (ECEEE)* (Lefke, Cyprus, November 2010), 118–123, https://doi.org/10.13140/2.1.4664.9288.

[189] M. Nabil, M. Aly, and A. Atiya, "LABR: A Large Scale Arabic Sentiment Analysis Benchmark," (2014), https://arxiv.org/abs/1411.6718.

[190] O. Einea, A. Elnagar, and R. Al Debsi, "SANAD: Single-Label Arabic News Articles Dataset for Automatic Text Categorization," *Data in Brief* 25 (2019): 104076, https://doi.org/10.1016/j.dib.2019.104076.

[191] I. A. El-khair, "1.5 Billion Words Arabic Corpus," (2016), https://arxiv.org/abs/1611.04033.

[192] T. Zerrouki and A. Balla, "Tashkeela: Novel Corpus of Arabic Vocalized Texts, Data for Auto-Diacritization Systems," *Data in Brief* 11 (2017): 147–151, https://doi.org/10.1016/j.dib.2017.01.011.

[193] Y. A. Alhaj and M. A. A. Al-qaness, "Feature Selection on Arabic Document Classification: Comparative Study Feature Selection on Arabic Document Classification: Comparative Study," *SAVE Proceedings* 15, no. ICIM (2018): 345–355.

[194] R. Suwaileh, M. Kutlu, N. Fathima, T. Elsayed, and M. Lease, "ArabicWeb16: A New Crawl for Today's Arabic Web," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (Pisa, Italy, July 2016), 673–676, https://doi.org/10.1145/2911451.2914677.

[195] A. Elnagar, L. Lulu, and O. Einea, "An Annotated Huge Dataset for Standard and Colloquial Arabic Reviews for Subjective Sentiment Analysis," *Procedia Computer Science* 142 (2018): 182–189, https://doi.org/10.1016/j.procs.2018.10.474.

[196] A. Elnagar, Y. S. Khalifa, and A. Einea, "Hotel Arabic-reviews Dataset Construction for Sentiment Analysis Applications," *Studies in Computational Intelligence* 740 (2018): 35–52, https://doi.org/10.1007/978-3-319-67056-0_3.

[197] E. Selab and A. Guessoum, "Building TALAA, a Free General and Categorized Arabic Corpus," *Proceedings of the International Conference on Agents and Artificial Intelligence* 1 (2015): 284–291, https://doi.org/10.5220/0005352102840291.

[198] Y. Altaher, A. Fadel, M. Alotaibi, et al., "Masader Plus: A New Interface for Exploring +500 Arabic NLP Datasets," (2022), https://arxiv.org/abs/2208.00932.

[199] "Intelligent Internet of Medical Things for Depression: CurrentAdvancements, Challenges, No Title".

[200] E. Liaras, M. Nerantzidis, and A. Alexandridis, "Machine Learning in Accounting and Finance Research: a Literature Review," *Review of Quantitative Finance and Accounting* 63, no. 4 (2024): 1431–1471, https://doi.org/10.1007/s11156-024-01306-z.

[201] W. Long, J. Gao, K. Bai, and Z. Lu, "A Hybrid Model for Stock Price Prediction Based on Multi-View Heterogeneous Data," *Financial Innovation* 10, no. 1 (2024): 48, https://doi.org/10.1186/s40854-023-00519-w.

[202] X. Zhuo, F. Irresberger, and D. Bostandzic, "How Are Texts Analyzed in Blockchain Research? A Systematic Literature Review," *Financial Innovation* 10, no. 1 (2024): 60, https://doi.org/10.1186/s40854-023-00501-6.

[203] F. Akhtar, J. P. Li, M. B. Bin Heyat, et al., "Potential of Blockchain Technology in Digital Currency: A Review," in *2019 16th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP 2019)* (Chengdu, China: IEEE, June 2019), 85–91, https://doi.org/10.1109/ICCWAMTIP47768.2019.9067546.

[204] M. Farahani, M. Gharachorloo, M. Farahani, and M. Manthouri, "ParsBERT: Transformer-Based Model for Persian Language Understanding," *Neural Processing Letters* 53, no. 6 (2021): 3831–3847, https://doi.org/10.1007/s11063-021-10528-4.

[205] A. Zafar, M. Wasim, S. Zulfiqar, T. Waheed, and A. Siddique, "Transformer-Based Topic Modeling for Urdu Translations of the Holy Quran," *ACM Transactions on Asian and Low-Resource Language Information Processing* 23, no. 10 (2024): 1–21, https://doi.org/10.1145/3694967.

[206] K. S. Esmaili, S. Salavati, and A. Datta, "Towards Kurdish Information Retrieval," *ACM Transactions on Asian Language Information Processing* 13, no. 2 (2014): 1–18, https://doi.org/10.1145/2556948.