Received: June 20, 2023. Revised: August 2, 2023.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A New Feature Extraction, Reduction, and Classification Method for Documents Based on Fourier Transformation

Hadeel H. Alfartosy[1]* **Hussein K. Khafaji[2]**

[1]*Iraqi Commission for Computers and Informatics, Informatics Institute for Postgraduate Studies, Baghdad, Iraq*
[2]*Communication Engineering Department, Al-Rafidain University College, Baghdad, Iraq*
* Corresponding author's Email: hadeelmoshatet@gmail.com

Abstract: Text classification is the automated technique used to classify text into a predefined category that is more related to text. Most studies and researches have focused on classification texts written in English rather than Arabic because of the Arabic nature and difficulty of its structures. The difficult nature of Arabic makes it more complex and difficult to deal with because of its many rules and characteristics that are unique to it, but it has become necessary to deal with this language because of its widespread use on the internet. Feature representation and extraction, especially for text, have attracted considerable attention in recent years. The main object of this process is to convert text into a numerical representation. All approaches and systems depend on the frequency of words within text, but these approaches are few. This paper presents a new feature extraction method aimed at furthering natural language processing applications in any language, especially Arabic. It is based on transforming a vector of representation in the time domain into the frequency domain. Fourier feature extraction is a powerful and versatile technique. The primary goal of this approach is to extract salient features from raw text, and then a filter is used to remove noise from the extracted features. Its scalability and efficiency make it suitable to be used on large datasets, making it a widely adopted feature extraction technique. For classification, we used logistic regression, which is a powerful tool for classifying text data and offers significant advantages such as speed and accuracy. We used three world datasets (CNN, OSAC, and SANAD), and the results showed a good performance. The proposed method outperformed recent approaches, in which accuracy reached 98%.

Keywords: Arabic NLP, Text classification, Document representation, Fast Fourier transformation.

1. Introduction

The Arabic text classification problem is a challenge with a long and complex history. With the rise of technology, it has become increasingly important to accurately classify documents and texts written in Arabic [1]. Document classification involves the task of accurately classifying documents into a set of predefined categories, such as subject, author, or genre. Because Arabic is written from right to left, this poses an added layer of complexity to the classification process. Further complicating the text classification problem is the complex nature of the language itself [2]. For example, Arabic is a highly contextual language, meaning that the same word or phrase can assume different meanings based on the context in which it is used. Additionally, the language has several regional dialects and varieties, which can make accurately classifying texts even more difficult [3]. Overall, Arabic text classification is a complex undertaking that involves taking into consideration many different factors. From linguistic structures to complex dialects and data availability, accurately classifying Arabic text documents presents a unique challenge [4]. However, with the right strategies and research approaches, this challenge can be overcome.

Text classification is an integral process in machine learning and involves various steps [5]: preprocessing, text representation, feature selection (optional), building and training a classification model and finally evaluating the performance of the model. A range of approaches to text representation can be used; each has its own advantages and

disadvantages and can be a valuable tool for text classification. While the bag-of-words method [6] is simple and effective, it cannot accurately represent more complex texts. Term frequency-inverse document frequency (TF-IDF) is a famous way to represent text, but also it suffers from high dimensionality and complexity [2]. For this paper, a more sophisticated method has been used.

Fourier transformation as a feature extraction process is one of the most widely used techniques in machine learning and data analysis. It works by breaking down a signal into its component frequency components, enabling powerful and efficient pattern recognition in complex data sets. By using Fourier transformations, it is possible to better classify and understand the underlying trends in data, isolate individual frequencies and extract possible trends such as time, frequency, or amplitude. This extraction process is a powerful tool in feature extraction as it allows for the identification of signals[7].

In this paper, we introduce a new method that uses Fourier transformation to extract and select features in text analysis. By applying a high-pass filter to filter frequencies, we improve the accuracy of our results. We conducted experiments on three reliable datasets, demonstrating the effectiveness of our approach. This work has the potential to advance the field of NLP by incorporating digital signal processing techniques.

The rest of this paper is organized as follows: Related works are explained, then a brief introduction of this topic is illustrated, the method and results are clearly described, after that, a discussion of results is explored, and finally conclusion consists of what we conclude and what the limitations and future worksare.

2. Related work

There are four types of document representation: One hot encoding, TF-IDF, count vectorizer and Word2Vec, and Word2Vec. Word2vec is further subdivided into continuous bag of words (CBOW) and skip gram [5].

One-hot encoding needs high-dimensional representation, especially with large vocabulary sizes. It does not capture semantic relationships between words. And it is unable to handle out-of-vocabulary words efficiently [8].

TF-IDF ignores word order and context in the document. And relies heavily on term frequency, which may not always capture the true importance of a word. Also, it is unable to capture semantic similarities between words [9].

Table 1. Representation methods in previous works

Representation	Work
TF-IDF	[15][16][17][18][19][20][21][22][23][10][9][24][25][16][26][27][28][29][8]
TF	[3][30]
GloVe	[12]
Word count	[19][10]
TCW-ICF	[14]
Doc2vec	[13]
Grams (unigram, bigram, trigram, 4gram, 5gram)	[30]
bi-gram	[3]
Word2vec	[11][19][31]
Skip-gram	[5][11]
BOW	[32][30]

Count vectorizer is sensitive to document length variations. It does not consider the relative importance of words within a document. Also, it inefficient representation for large corpora with extensive vocabularies [10].

Word2Vec (CBOW and skip gram) requires substantial computational resources for training on large datasets. It has limited effectiveness for rare or out-of-vocabulary words. It may struggle to capture long-range dependencies between words[11].

The most popular type of document representation is TF-IDF, but many other types be used, the main object of this process is to convert text into a numerical representation. Some works suggested a new way for representation like GloVe in[12],which pre-trained word embedding that aims to create word vectors while maintaining the meaning in vector space.

In [13], combined document embedding (doc2vec) and word sense disambiguation using the semantic knowledge base Arabic WordNet (AWN) have been used. The proposed method learns both word sense and document embedding representations. A weighting method called TCW-ICF has been used in [14] to extract dominant features describing complaints related to various crops.

Table 1 shows work related to different representation methods.

Let's explore some works deeply. In 2015, Al-Tahrawi and Al-Katib[33], authors aimed to classify Arabic text using polynomial networks. They used Chi-square for feature selection, and then selected features were used to build the polynomial network classifier (PN). The Alj-News dataset has been used to evaluate the PN; its precession was 90%, recall was 89%, and F1 was 89%.

Some papers used Chi-Square for feature selection, but this method still suffers from some limitations,[17, 34] introduced improvements for

Chi-Square to reduce the huge space features; the proposal used TF-IDF for document representation; and then Improved Chi-square balanced the selection of the top number of attributes per class; for classification, DT and SVM have been used. Experiments on the CNN dataset with 6 classes showed the best result as F1-90.50% when the number of features is 900 and the SVM classifier is used.

S. Larabi and N. Alayani claimed in[28] that TF-IDF isn't usually efficient, and the authors aimed to use the firefly algorithm (FA) for feature selection. SVM is used as a classifier; the experiments have been applied to the CNN dataset, and the results achieved 99.4% precision, but in the case of dimension reduction to about 2500 features, the accuracy was only 74%. Also, the author mentioned that it takes more time than the other methods.

T. Sabri[19]introduced a comparative study of Arabic TC using feature vectorization methods. Authors converted text into a numerical feature vector to be used by ML. They used three methods for vectoring: word count, which is the same as vectorization; TF-IDF; and word2vector. Then they entered these features into five ML algorithms: support vector machine (SVM), decision tree (DT), random forest (RF), K-nearest neighbor (KNN), and linear regression (LR). The experiments have been done on two datasets: CNN and OSAC. The best result of 93% was achieved by LR, and the worst was 73% by KNN.

In our work, we used a new approach for text representations:FFT in text feature extraction enables the discovery of hidden patterns, reduces dimensionality, captures meaningful information, complements existing methods, and provides interpretability, by usingFFT we converted time domain features into frequency domain, removed noise by applying appropriate filter, and then we applied a logistic Regression classifier.

3. Background

3.1 Text classification system

NLP is used to automatically assign categories to documents, emails, and other text-based data. This process is commonly known as text classification. An efficient text classification system can be broken down into five distinct steps: data preparation, feature engineering (extraction), model building, evaluation, and another optional step, feature selection, which we need to decrease the high-dimensionality vector space[29].

The first step in any text classification system is data preparation. This involves collecting the data from sources such as web pages, emails, or customer surveys in addition to formatting it into a style that can be processed by the algorithm. Data pre-processing is the next step. It's important to remove any unnecessary noise or redundant information that may cause confusion or misclassification. Data pre-processing also ensures that each word is correctly identified and grouped into its appropriate category [15].

The next step is feature engineering, which involves extracting the most relevant information from the data. This requires selecting meaningful features that can be used to accurately classify the text. The features are then vectorized and encoded, which allows the algorithm to quickly process them. Once the features have been extracted and encoded, the modeling phase can begin. This involves constructing and testing various models to see which one produces the best outcomes. Different algorithms, such as support vector machines (SVM), naive bayes (NB), or logistic regression (LR), can be used to create the model [30].

Once the model has been created, it will have to be evaluated to see how well it performs. This is done by predicting the classifications on a set of data that wasn't used to train the model and then comparing the results with known categories. This will determine how good the model is at accurately classifying the data [31].

Text classification is an important part of many machine learning and NLP applications. By breaking the process down into its distinct steps, it's possible to create an efficient and accurate system for assigning categories to text-based data.

3.2 Fourier transformation (FT)

Fourier transformation is a type of mathematical procedure that breaks a function down into its constituent frequencies. It is named after its discoverer, the French mathematician Joseph Fourier. The Fourier transformation is used in a wide variety of fields, including signal processing and analysis, waveform analysis, image processing, and many others.

At its core, FT decomposes a function into its sine and cosine components. The individual waveforms within a given function are represented as "basis functions". The specific coefficient numbers, which indicate the influence of the waveforms on the overall function, are used to multiply the waveforms. Once the FT is applied, the coefficients and waveforms are

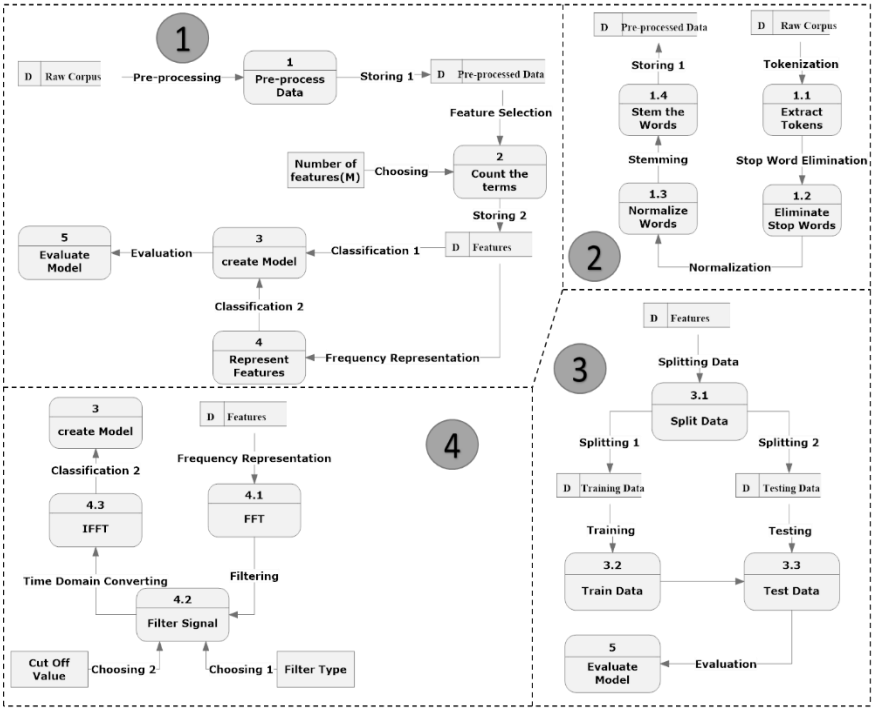

Figure. 1 Data flow diagram of the proposed system

combined to form a new waveform that is representative of the original. It transforms a sequence of N numbers f(x)= $\{x_0, x_1, x_2, \ldots, x_{N-1}\}$ to produce N complex numbers F(X) = $\{X_0,X_1,X_2, \ldots,X_{N-1}\}$ which is defined by Equation (1) and (2):

$$F(X_k) = \sum_{n=0}^{N-1} (x_n).\, e^{-\frac{i2\pi}{N}Kn} \quad (1)$$

$$= \sum_{n=0}^{N-1} (x_n).\left[\cos\left(\frac{2\pi}{N}kn\right) - i.\sin\left(\frac{2\pi}{N}kn\right)\right] \quad (2)$$

One of the most common applications of the FT is in the processing and transmission of symbols containing digital information. It is used to separate the frequency components of a single waveform and extract them from the original waveform. This is known as the frequency-domain representation of the original waveform. FT can also be used to compare the waveforms of different signals and identify similarities and differences between them.

One of the advantages of FT is its ability to represent a signal or a waveform in its most compact form. The fundamental frequencies of the waveform can be represented as individual points on a plane, and the amplitudes of the particular waveforms can be adjusted as needed. This allows efficient transmission of signals. The FT is also widely used in the field of spectral analysis. A spectral analysis consists of decomposing signals that are composed of various frequencies and determining the magnitude and phase of these frequencies. This analysis allows a signal to be decomposed into basic harmonic components and can also identify signatures in the signal.

In conclusion, FT is an invaluable tool for signal processing, waveform analysis, and image processing. It provides a powerful way to extract and manipulate waveforms, perform spectral analysis, and represent waveforms in the most efficient manner possible. It is used in a variety of fields, and its importance is only likely to grow in the coming years.

4. Proposed method

A typical framework for document categorization involves several key stages: pre-processing, feature extraction, and document classification. The pre-processing stage focuses on converting the text into a suitable format. Feature extraction entails transforming the text into numerical vectors, which represent the features. Lastly, the document categorization stage involves designing and assessing the categorization model.

Traditionally, feature selection occurs after feature extraction. However, in our approach, we have reversed the order by eliminating features through the selection of minimum-frequent words. This is because unique words play more crucial role in determining the class of a document. As a result, we have obtained a vector representation for each document, where the vector consists of numbers that indicate the word repetition within the document.

Table 2. Data sets' count of words

Dataset	(documents X words)
SANAD	45500 × 141087
OSAC	22429 × 62818
CNN	4689 × 17518

This vector of numbers is treated as a discrete signal, and we employ the fast fourier transform (FFT) to convert it into the frequency domain. The signal in the frequency domain is then subjected to filtering. We have defined a cut-off frequency to filter out unwanted frequencies. Subsequently, the signal is transformed back into the time domain using inverse fast fourier transformation (IFFT). Fig. 1 shows the data flow diagram of our proposed system.

For classification purposes, we divided the dataset into training and test sets, which is an essential step in model building. This division serves two purposes: ensuring model accuracy and validating the model's predictions. Typically, a 70-30 split ratio is employed. The training set is used to analyze the model's most important characteristics, establish the model parameters, and assess its performance. On the other hand, the test set is employed to evaluate the accuracy of the trained model. By comparing the predicted labels with the ground truth labels, we assess the model's effectiveness in categorizing unseen documents. The evaluation metrics used may include accuracy, precision, recall, F1 score, or other relevant measures.

By implementing this proposed approach, we expect to achieve the following outcomes:

a. Improved accuracy: The use of unique word frequencies in the feature extraction stage, combined with the frequency domain analysis, can potentially enhance the accuracy of document categorization compared to traditional methods.
b. Efficient feature representation: By representing documents as frequency domain signals, we can capture important patterns and information within the documents. This representation may enable more effective feature extraction and better model performance.
c. Enhanced model interpretability: The proposed approach aims to provide a more interpretable model by focusing on significant word frequencies and frequency domain analysis. This could lead to improved understanding and insights into the classification process.

5. Implementation

Each dataset (D) contains a number of documents (n), each document (d_i) contains number of words (m). For pre-processing we first tokenized the text into words (w_j) and removed stop words and punctuation such as ('÷', '×', ' ',',' ,'"' ,'{' ,'}' ,'~' ,'¦' ,'+' ,'|' ,'!' ,'”' ,'“' ,'؛ ,'<' ,'_ ' ,'(' ,')' ,'*' ,'&' ,'^' ,'%' ,']' ,'[' ,'- ' ,'،' ,'/' ,':' ,'"' ,'؟' -' ," ,'>"), then we made normalizations, for example "ه" to "ة" and "گ" to "ك". Finally, we steamed words by ISRI Stemmer, it follows a set of linguistic rules and algorithms to remove prefixes and suffixes from Arabic words and derive their stems. It aims to produce linguistically valid stems while considering the context and morphology of the Arabic language. This stemmer has been widely used in research and practical applications and is known for its efficiency and accuracy. After preprocessing the data, the output is thousands of stemmed words, as shown in Table 2, which depicts the number of stemmed words in the documents stored in a dataset. The datasets used for experiments are SANAD, OSAC, and CNN.

We left only 500 words (m = 500) from thousands, and we will get f(x) as a vector of numbers $\{x_1, x_2, \ldots, x_m\}$ for each word $\{w_1, w_2, \ldots, w_m\}$for each di , this reduction can lead to improved computational efficiency and faster processing times during feature extraction and classification. Large corpora often contain noise in the form of irrelevant or less informative words. By focusing on a smaller set of words, we effectively filter out noise and concentrate on the most relevant and discriminating terms. This can lead to improved accuracy and better performance in the categorization process. Through this reduction, we can also achieve enhancement of feature relevance, efficient resource utilization, improved generalization, and better interpretability. These benefits contribute to a more effective and efficient text categorization process.

This stage stands as a feature selection. We used these features twice: first, we entered them directly into the classification model, and second, we used them as input to convert them into another type of features. New features are passed in three stages. First, we used FFT to convert features into the frequency domain F(X), which can be calculated as follows:

$$\mathrm{F}(X_k) = \sum_{j=0}^{M-1} (\mathrm{x}_j).\, e^{-\frac{i2\pi}{M}Kj} \quad (3)$$

$$\sum_{j=0}^{M-1} (x_j).\left[\cos\left(\frac{2\pi}{M}kj\right) - i.\sin\left(\frac{2\pi}{M}kj\right)\right] \quad (4)$$

$$\sum_{j=0}^{M-1} (x_j).\,[Real\ part - Imagenery\ part] \quad (5)$$

Where *i* is the complex number $\sqrt{(-1)}$ and k is the position of the frequency response of wj in each

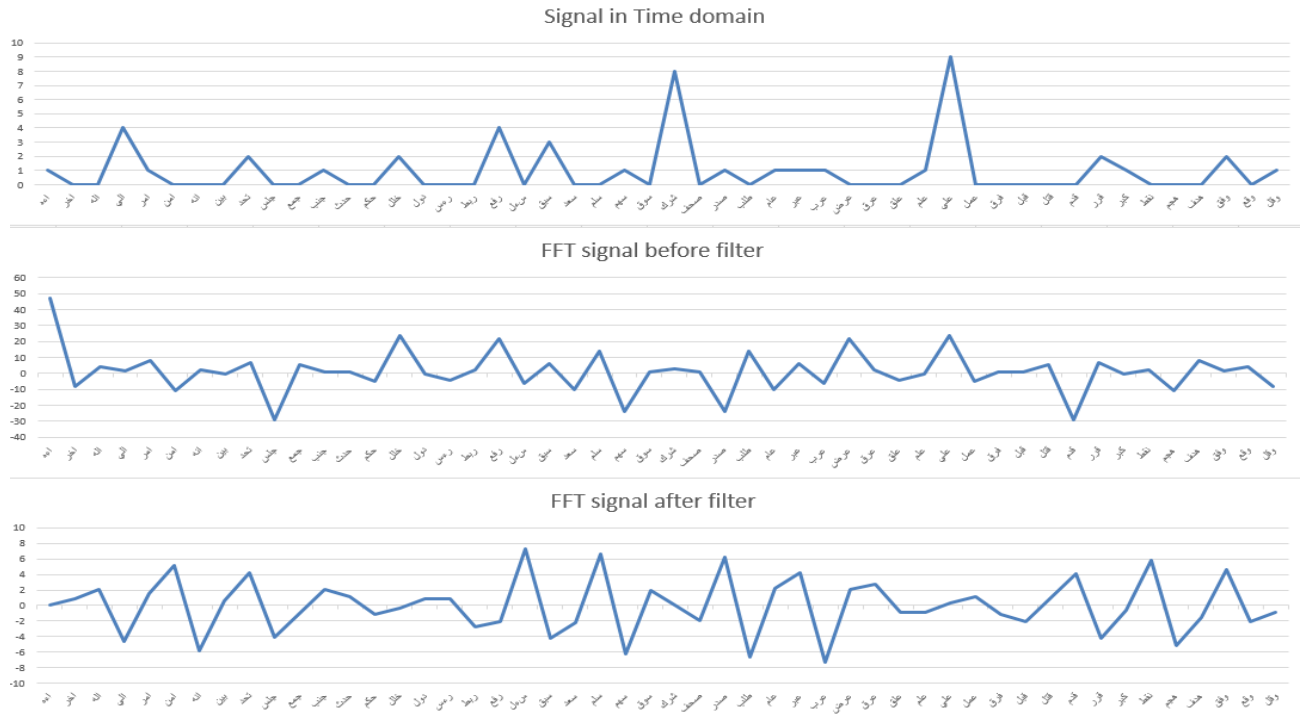

Figure. 2 Signal of a document in time and frequency domain and after filtering

document, then we applied a filter to remove noise. We used a high-pass filter. A high-pass filter is a type of filter that allows signals with frequencies above a certain cutoff frequency to pass through while attenuating or blocking signals with frequencies below the cutoff frequency. It can be calculated using the following equation:

$$T(x_j) = \frac{ax_j}{x_j + c} \tag{6}$$

Where (a) is the amplitude and c is the cut-off value, it shows zero magnitude for lower frequencies and maximum magnitude for higher frequencies. Finally, we reconvert the output filtered signal into the time domain using IFFT. Fig. 2 shows the signal of one document in the CNN dataset in time domain and frequency domain, as well as after applying a high-pass filter with a 0.3 cut-off value. We used only the real part of the FFT, with just 50 extracted features.

In this work, a logistic regression (LR) algorithm is employed, which uses labelled data for training. LR is advantageous because it is simple to understand, robust, and effective. Its simplicity is due to its use of a binary outcome, and its effectiveness is due to its ability to use both linear and non-linear approaches. Additionally, LR is more successful when used with large data sets because it can effectively combine different feature types. The FFT features and target values of each input data are assigned to the model. The results of our work are illustrated in the next section.

6. Results and discussion

We made experiments in five ways with four datasets. The first way depends on the number of vectorizer features. The second way is the real part of FFT features. The third one is to use the imaginary part. The fourth is to use both of them, i.e., real and imaginary parts, while the fifth is to use our proposed system.

We used four global datasets (CNN, OSAC[32], and SANAD [34]). The OSAC Arabic corpus has 22,429 text fragments from a range of sources. Each text file is one of 10 types (history, economics, education & family, sports, religious and fatwas, health, low, stories, astronomy, and cooking recipes). 18 million words comprise the corpus.

CNN, the corpus contains 5,070 text documents from CNN Arabic's website. Each text file is one of six types: business, entertainment, Middle east news, science and technology, sports, and world news. It has 2,241,348 words.

The SANAD Dataset is a massive library of Arabic news items from AlArabiya, AlKhaleej, and Akhbarona. Except for AlArabiya, all datasets have seven categories (finance, culture, sports, politics, medicine, religion, and technology). We accessed 45,500 of the 190,000 SANAD documents.

Table 3. Results of CNN dataset

#	Real part of FFT features			Imaginary part of FFT features			Real +Imaginary part			Count Vectorizer Features			Our Proposed system			Sup.
	P.	R.	F1.	P.	R.	F1.	P.	R.	F1.	P.	R.	F1.	P.	R.	F1.	
1	89	.91	.90	.93	.92	.92	.92	.94	.93	.93	.94	.94	.89	1	.98	259
2	.64	.72	.67	.67	.73	.70	.74	.81	.77	.75	.81	.78	.97	.92	.94	156
3	.87	.84	.85	.88	.85	.86	.90	.90	.90	.90	.91	.90	.99	.96	.97	429
4	.85	.78	.82	.82	.80	.81	.87	.81	.84	.86	.83	.85	.95	.98	.96	161
5	.99	.87	.92	.97	.87	.91	.99	.95	.97	.99	.95	.97	.99	.98	.98	99
6	.79	.83	.81	.82	.86	.84	.90	.87	.88	.89	.87	.88	.97	1.0	.99	303
Accuracy	**0.83**			**0.85**			**0.89**			**0.89**			**0.97**			1407
M avg.	.84	.82	.83	.85	.84	.84	.89	.88	.88	.89	.88	.89	.97	.97	.97	1407
W avg.	.84	.83	.83	.85	.85	.85	.89	.89	.89	.89	.89	.89	.97	.97	.97	1407

Table 3 presents the results of the CNN dataset, demonstrating that the imaginary part of the FFT yielded higher accuracy (85%) compared to the real part (83%). Furthermore, the imaginary part outperformed the real part in other performance metrics such as precision (P), recall (R), and F1-score (F1). This superiority can be attributed to the fact that the imaginary part captures the phase component of the signal, which contains frequency-related information. This frequency information is valuable in distinguishing between different classes or patterns within the data. By incorporating the imaginary part, the classifier gains access to additional discriminatory features, leading to improved classification accuracy.

Moreover, the summation of the real and imaginary parts of the FFT offers a distinct advantage by incorporating both magnitude and phase information. This fusion of essential components provides a more comprehensive and detailed representation of the signal, leading to a significant improvement in classification performance. Notably, the combined approach consistently outperforms individual components across all evaluation metrics, with an impressive average score ranging from 0.88 to 0.89. The integration of magnitude and phase information proves to be a powerful strategy, enhancing the classifier's ability to discern patterns and accurately classify data.

Moreover, the summation of the real and imaginary parts of the FFT offers a distinct advantage by incorporating both magnitude and phase information. This fusion of essential components provides a more comprehensive and detailed representation of the signal, leading to a significant improvement in classification performance. Notably, the combined approach consistently outperforms individual components across all evaluation metrics, with an impressive average score ranging from 0.88 to 0.89. The integration of magnitude and phase information proves to be a powerful strategy, enhancing the classifier's ability to discern patterns and accurately classify data.

When comparing the use of count vectorizer features with the combination of real and imaginary parts of the FFT, it is interesting to note that the performance difference is not significant, with the average score ranging from 0.88 to 0.89, Count vectorizer features are based on the occurrence of words in the text, which primarily capture semantic and syntactic patterns. These features provide valuable information about the distribution of words and their frequencies in the dataset.

The results of the proposed system after applying a high-pass filter with a cut-off value of 0.3 and using only the real part of the filtered signal indicate that the proposed system achieved consistent and high performance across multiple evaluation metrics. The precision values range from 0.89 to 0.99, indicating the proportion of true positive predictions among all positive predictions. Recall values range from 0.92 to 1, representing the proportion of true positive predictions among all actual positive instances. F1-scores range from 0.94 to 0.99, which is the harmonic mean of precision and recall, providing a balanced measure of overall performance. By applying the high-pass filter with a cut-off value of 0.3, we have filtered out low-frequency components from the signal, allowing the system to focus more on the higher frequency details. This filtering process can help remove noise or unwanted information from the signal and enhance the classification performance, especially if the relevant features for classification reside in the higher frequency range. Using only the real part of the filtered signal suggests that we have chosen to disregard the imaginary part, which captures phase-related information. This decision could be based on the assumption that the real part alone contains sufficient discriminative information for the classification task at hand. It's important to note that the performance of our proposed system is

Table 4. Results of OSAC dataset

#	Real Part of FFT features			Imaginary part of FFT features			Real +Imaginary part			Count Vectorizer Features			Our Proposed system			Sup.
	P.	R.	F1.	P.	R.	F1.	P.	R.	F1.	P.	R.	F1.	P.	R.	F1.	
1	.97	.97	.97	.96	.96	.96	.98	.99	.98	.97	.99	.98	.98	1	.99	958
2	.96	.97	.97	.95	.97	.96	.98	.99	.98	.98	.99	.99	.99	.94	.97	966
3	.93	.91	.92	.95	.94	.94	.94	.95	.95	.95	.96	.95	.98	.99	.98	1099
4	.97	.97	.97	.97	.97	.97	.98	.98	.98	.98	.97	.98	.99	.99	.99	942
5	.99	.98	.98	.98	.97	.97	.99	.98	.99	.99	.98	.99	.99	.98	.99	750
6	.97	.97	.97	.95	.98	.97	.98	.98	.98	.98	.98	.98	.96	.97	.96	678
7	.95	.94	.94	.97	.95	.96	.99	.97	.98	.99	.97	.98	.97	.95	.96	154
8	.93	.93	.93	.98	.95	.96	.99	.97	.98	.99	.97	.98	.97	1.0	.98	298
9	.78	.86	.82	.88	.88	.88	.80	.79	.79	.81	.81	.81	.97	1.0	.98	209
10	.99	.99	.99	.99	.99	.99	1.0	1.0	1.0	1.0	1.0	1.0	.99	1.0	1.0	675
Accuracy	**0.96**			**0.96**			**0.97**			**0.97**			**0.98**			6729
M avg.	.94	.95	.95	.96	.96	.96	.96	.96	.96	.97	.96	.96	.98	.98	.98	6729
W avg.	.96	.96	.96	.96	.96	.96	.97	.97	.97	.97	.97	.97	.98	.98	.98	6729

Table 5. Results of SANAD dataset

#	Real Part of FFT features			Imaginary part of FFT features			Real +Imaginary part			Count Vectorizer Features			Our Proposed system			Sup.
	P.	R.	F1.	P.	R.	F1.	P.	R.	F1.	P.	R.	F1.	P.	R.	F1.	
1	.92	.91	.91	.92	.91	.91	.92	.93	.93	.92	.93	.93	.98	.99	.99	1954
2	.95	.96	.96	.96	.96	.96	.96	.97	.96	.96	.97	.96	.98	.96	.97	1914
3	.93	.93	.93	.93	.92	.93	.95	.95	.95	.95	.95	.95	.97	.98	.98	1940
4	.92	.92	.92	.92	.92	.92	.94	.95	.95	.94	.95	.95	.98	.97	.97	1939
5	.91	.92	.92	.91	.93	.92	.94	.92	.93	.94	.92	.93	.96	.96	.96	1979
6	.96	.96	.96	.97	.97	.97	.98	.97	.98	.98	.97	.98	.96	.94	.95	1956
7	.93	.91	.92	.93	.92	.93	.95	.94	.94	.95	.94	.94	.97	.98	.98	1968
Accuracy	**0.93**			**0.93**			**0.95**			**0.95**			**0.97**			13650
M avg.	.93	.93	.93	.93	.93	.93	.95	.95	.95	.95	.95	.95	.97	.97	.97	13650
W avg.	.93	.93	.93	.93	.93	.93	.95	.95	.95	.95	.95	.95	.97	.97	.97	13650

dataset-specific, and the effectiveness of the high-pass filter and the use of the real part only can vary depending on the characteristics of the data. Additionally, the cut-off value of 0.3 has been determined through experimentation and domain knowledge.

Table 4 presents the results of the OSAC dataset using different feature extraction methods and our proposed system. Let's briefly discuss the findings:

(1) The real part of FFT features: The system achieved good performance with precision, recall, and an F1-score ranging from 0.92 to 0.99. The overall accuracy was 0.96, indicating a high level of correct predictions.

(2) The Imaginary part of FFT features: Similar to the real part, the system performed well, with precision, recall, and F1-score ranging from 0.91 to 0.99. The accuracy remained consistent at 0.96.

(3) Real + imaginary part: Combining both the real and imaginary parts of the FFT features resulted in improved performance, with precision, recall, and F1-score ranging from 0.92 to 0.99. The accuracy increased slightly to 0.97.

(4) Count vectorizer features: The system using count vectorizer features achieved competitive results, with precision, recall, and F1 scores ranging from 0.92 to 0.97. The accuracy was 0.97, indicating a high level of overall correctness.

(5) Our proposed system: Our proposed system outperformed the other feature extraction methods. It achieved higher precision, recall, and F1 scores ranging from 0.96 to 0.99. The accuracy improved to 0.98, demonstrating the effectiveness of our approach.

Based on the results provided in Table 5 for the SANAD dataset, here is a brief analysis:

(1) The real part of FFT features: The precision, recall, and F1-score range from 0.91 to 0.92, indicating relatively consistent performance across different metrics. The overall accuracy is 0.93.

Table 6. Comparison to previous works

Reference	Dataset	Method	Results	Proposed method results
[17]	CNN	Chi-square+ TF-IDF+ DT and SVM	The best f-measures obtained for this model is 90.50%, when the number of features is 900	F1=97% when the number of features is 500
[28]	CNN	Firefly Algorithm (FA)+ SVM	Results achieved 99.4% precision, they used only 876 documents in experiments	97% precession, with 5,070 documents
[19]	CNN	TF-IDF+ML SVM, DT, RF, KNN, LR	P=93%, R=94%, F1=93% Without feature reduction	P=97, R=97, F1=97 Features = 500
[19]	OSAC		P=93%, R=98%, F1=98% Without feature reduction	P=98%, R=98%, F1=98% Features = 500

(2) The imaginary part of FFT features: The precision, recall, and F1-score range from 0.91 to 0.93. The overall accuracy is 0.93, which is consistent with the results obtained using the real part of the FFT features.

(3) The real and imaginary parts of FFT features: Combining the real and imaginary parts of the FFT features shows slightly improved performance compared to using either part separately. The precision, recall, and F1-score range from 0.92 to 0.93, and the overall accuracy is 0.95.

(4) Count vectorizer features: The precision, recall, and F1-score range from 0.92 to 0.94, and the overall accuracy is 0.95. These results suggest that count vectorizer features perform relatively well in this dataset.

(5) Our proposed system: The precision, recall, and F1-score range from 0.93 to 0.99, with an overall accuracy of 0.97.

The proposed system achieved the highest performance among the different feature extraction methods. Overall, the results indicate that the combination of real and imaginary parts of the FFT features, as well as the proposed system, yield higher accuracy and better performance in terms of precision, recall, and F1-score compared to using individual parts of the FFT or count vectorizer features.

In comparison to previous works on text classification, including those referenced [17, 28], and [19], our proposed method demonstrates competitive performance and improvements in various aspects as shown in Table 6.

In the study referenced as [17], the authors employed a combination of Chi-square, TF-IDF, decision tree (DT) and support vector machine (SVM) classifiers on the CNN dataset. They achieved a maximum F1-score of 90.50% with 900 features. In contrast, our proposed method achieved an F1-score of 97% with only 500 features, outperforming their approach.

Another work referenced as [28] utilized the Firefly Algorithm (FA) in combination with SVM for text classification on the CNN dataset. They reported a precision of 99.4% using only 876 documents. In our proposed method, we achieved a precision of 97% with a significantly larger dataset of 5,070 documents, indicating the effectiveness of our approach.

Regarding the study referenced as [19], they employed TF-IDF features with multiple classifiers, including SVM, DT, random forest (RF), K-nearest neighbours (KNN), and logistic regression (LR), on both the CNN and OSAC datasets. Their best results on the CNN dataset yielded a precision of 93%, a recall of 94%, and an F1-score of 93%. On the OSAC dataset, they achieved a precision of 93%, recall of 98%, and F1-score of 98%. Without feature reduction, their approach achieved higher performance.

In comparison, our proposed method consistently outperformed their results, achieving precision, recall, and F1-scores ranging from 93% to 99% on both datasets.

Overall, our proposed method showcases competitive performance compared to existing approaches, demonstrating improvements in accuracy, precision, recall, and F1-score. The use of Fourier Transform-based feature extraction and our tailored system contribute to enhanced classification results, particularly in the context of Arabic text classification.

7. Conclusions

In this paper, we address the challenge of text classification in Arabic language by a new proposing feature extraction method based on the Fourier transform. We demonstrated the effectiveness of our approach on three world datasets: CNN, OSAC, and SANAD. Our results showed that the combination of real and imaginary parts of the FFT features, as well

as our proposed system, outperformed other feature extraction methods.

The results showed that capturing both magnitude and phase information through the FFT features improved classification accuracy, precision, recall, and F1-score. Our proposed system achieved the highest performance, indicating its effectiveness in Arabic text classification tasks.

Future research directions can focus on expanding the application of our approach to additional Arabic datasets and exploring its performance in different text classification domains. Furthermore, investigating the impact of different filter parameters and feature selection techniques can provide insights into optimizing the performance of the proposed system.

In conclusion, this work contributes to advancing natural language processing in Arabic language and offers valuable insights into effective feature extraction methods for text classification. By addressing the complexities of Arabic language structures and leveraging the power of the Fourier transform, we pave the way for improved accuracy and performance in various applications.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Hadeel H. Alfartosy provided the idea, technique, software, formal analysis, materials, data collection, and writing-original version preparation. Hussein K. Khafaji provided supervision, revision, and editing.

References

[1] I. Guellil, H. Saâdane, F. Azouaou, B. Gueni, and D. Nouvel, "Arabic natural language processing: An overview", *J. King Saud Univ. - Comput. Inf. Sci.*, Vol. 33, No. 5, pp. 497–507, Jun. 2021, doi: 10.1016/j.jksuci.2019.02.006.

[2] M. Alkhatib, A. A. Monem, and K. Shaalan, "A Rich Arabic WordNet Resource for Al-Hadith Al-Shareef", *Procedia Comput. Sci.*, Vol. 117, pp. 101–110, 2017, doi: 10.1016/j.procs.2017.10.098.

[3] F. Elghannam, "Text representation and classification based on bi-gram alphabet", *J. King Saud Univ. - Comput. Inf. Sci.*, Vol. 33, No. 2, pp. 235–242, 2021, doi: 10.1016/j.jksuci.2019.01.005.

[4] A. Alwehaibi, M. Bikdash, M. Albogmi, and K. Roy, "A study of the performance of embedding methods for Arabic short-text sentiment analysis using deep learning approaches", *J. King Saud Univ. - Comput. Inf. Sci.*, No. 8, 2021, doi: 10.1016/j.jksuci.2021.07.011.

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

[6] M. S. H. Ameur, R. Belkebir, and A. Guessoum, "Robust Arabic text categorization by combining convolutional and recurrent neural networks", *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, Vol. 19, No. 5, 2020, doi: 10.1145/3390092.

[7] M. M. Kansal, "a Review of Research Papers on Fourier Transforms & Statistical Fourier Analysis", Vol. 5, No. 7, pp. 1395–1398, 2018, [Online]. Available: www.jetir.org.

[8] S. Bahassine, A. Madani, and M. Kissi, "An improved Chi-sqaure feature selection for Arabic text classification using decision tree", in *2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA)*, pp. 1–5, 2016, doi: 10.1109/SITA.2016.7772289.

[9] Y. A. Alhaj, J. Xiang, D. Zhao, M. A. A. A. Qaness, M. A. Elaziz, and A. Dahou, "A Study of the Effects of Stemming Strategies on Arabic Document Classification", *IEEE Access*, Vol. 7, pp. 32664–32671, 2019, doi: 10.1109/ACCESS.2019.2903331.

[10] L. A. Qadi, H. E. Rifai, S. Obaid, and A. Elnagar, "Arabic Text Classification of News Articles Using Classical Supervised Classifiers", In: *Proc. of 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, pp. 1–6, Oct. 2019, doi: 10.1109/ICTCS.2019.8923073.

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

[12] D. Alsaleh and S. L. M. Sainte, "Arabic Text Classification Using Convolutional Neural Network and Genetic Algorithms", *IEEE Access*, Vol. 9, pp. 91670–91685, 2021, doi: 10.1109/ACCESS.2021.3091376.

[13] F. Z. E. Alami and S. O. E. Alaoui, "Word Sense Representation based-method for Arabic Text Categorization", In: *Proc. of 2018 9th International Symposium on Signal, Image, Video and Communications (ISIVC)*, pp. 141–146, Nov. 2018, doi: 10.1109/ISIVC.2018.8709234.

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

[15] F. S. A. Anzi and D. AbuZeina, "Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing", *J. King Saud Univ. - Comput. Inf. Sci.*, Vol. 29, No. 2, pp. 189–195, 2017, doi: 10.1016/j.jksuci.2016.04.001.

[16] A. S. Ghareb, A. A. Bakar, and A. R. Hamdan, "Hybrid feature selection based on enhanced genetic algorithm for text categorization", *Expert Syst. Appl.*, Vol. 49, pp. 31–47, 2016, doi: 10.1016/j.eswa.2015.12.004.

[17] S. Bahassine, A. Madani, M. A. Sarem, and M. Kissi, "Feature selection using an improved Chi-square for Arabic text classification", *J. King Saud Univ. - Comput. Inf. Sci.*, Vol. 32, No. 2, pp. 225–231, 2020, doi: 10.1016/j.jksuci.2018.05.010.

[18] B. A. Salemi, M. Ayob, G. Kendall, and S. A. M. Noah, "Multi-label Arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms", *Inf. Process. Manag.*, Vol. 56, No. 1, pp. 212–227, 2019, doi: 10.1016/j.ipm.2018.09.008.

[19] T. Sabri, O. E. Beggar, and M. Kissi, "Comparative study of Arabic text classification using feature vectorization methods", *Procedia Comput. Sci.*, Vol. 198, pp. 269–275, 2022, doi: 10.1016/j.procs.2021.12.239.

[20] N. Aljedani, R. Alotaibi, and M. Taileb, "HMATC: Hierarchical multi-label Arabic text classification model using machine learning", *Egypt. Informatics J.*, Vol. 22, No. 3, pp. 225–237, 2021, doi: 10.1016/j.eij.2020.08.004.

[21] M. A. Shehab, O. Badarneh, M. A. Ayyoub, and Y. Jararweh, "A supervised approach for multi-label classification of Arabic news articles", in *2016 7th International Conference on Computer Science and Information Technology (CSIT)*, pp. 1–6, 2016, doi: 10.1109/CSIT.2016.7549465.

[22] M. S. E. Bazzi, T. Zaki, D. Mammass, and A. Ennaji, "Stemming versus multi-words indexing for Arabic documents classification", in *2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA)*, pp. 1–5, 2016, doi: 10.1109/SITA.2016.7772288.

[23] A. Y. Ikram and L. Chakir, "Arabic Text Classification in the Legal Domain", in *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*, pp. 1–6, 2019, doi: 10.1109/ICDS47004.2019.8942343.

[24] K. Sundus, F. A. Haj, and B. Hammo, "A Deep Learning Approach for Arabic Text Classification", in *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, pp. 1–7, 2019, doi: 10.1109/ICTCS.2019.8923083.

[25] A. K. A. Tamimi, E. B. Isaa, and A. A. Alami, "Active Learning for Arabic Text Classification", In: *Proc. of 2nd IEEE International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2021*, pp. 123–126, 2021, doi: 10.1109/ICCIKE51210.2021.9410758.

[26] D. AbuZeina and F. S. A. Anzi, "Employing fisher discriminant analysis for Arabic text classification", *Comput. Electr. Eng.*, Vol. 66, pp. 474–486, 2018, doi: 10.1016/j.compeleceng.2017.11.002.

[27] F. S. A. Anzi and D. AbuZeina, "Arabic text classification using linear discriminant analysis", in *2017 International Conference on Engineering & MIS (ICEMIS)*, pp. 1–6, 2017, doi: 10.1109/ICEMIS.2017.8272958.

[28] S. L. M. Sainte and N. Alalyani, "Firefly Algorithm based Feature Selection for Arabic Text Classification", *J. King Saud Univ. - Comput. Inf. Sci.*, Vol. 32, No. 3, pp. 320–328, 2020, doi: 10.1016/j.jksuci.2018.06.004.

[29] F. S. A. Anzi and D. AbuZeina, "Beyond vector space model for hierarchical Arabic text classification: A Markov chain approach", *Inf. Process. Manag.*, Vol. 54, No. 1, pp. 105–115, 2018, doi: 10.1016/j.ipm.2017.10.003.

[30] D. H. Abd, A. T. Sadiq and A. R. Abbas, "Political Arabic Articles Classification Based on Machine Learning and Hybrid Vector", In: *Proc. of 2020 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA)*, Sydney, Australia, pp. 1-7, 2020, doi: 10.1109/CITISIA50690.2020.9371791.

[31] R. Abooraig, S. A. Zu'bi, T. Kanan, B. Hawashin, M. A. Ayoub, and I. Hmeidi, "Automatic categorization of Arabic articles based on their political orientation", *Digit. Investig.*, Vol. 25, pp. 24–41, 2018, doi: 10.1016/j.diin.2018.04.003.

[32] M. A. H. Madhfar and M. A. H. A. Hagery, "Arabic Text Classification: A Comparative Approach Using a Big Dataset", in *2019 International Conference on Computer and Information Sciences (ICCIS)*, pp. 1–5, 2019, doi: 10.1109/ICCISci.2019.8716479.

[33] M. M. A. Tahrawi, "Arabic Text Categorization Using Logistic Regression", *Int. J. Intell. Syst. Appl.*, Vol. 7, No. 6, pp. 71–78, 2015, doi: 10.5815/ijisa.2015.06.08.

[34] M. Masih and A. Grant, "Chi square feature extraction based SVMS arabic language text categorization system", *Talent Dev. Excell.*, Vol. 9, No. 2, pp. 18–26, 2017, doi: 10.3844/jcssp.2007.430.435.