applied sciences

MDPI

Article

Text Classification of Patient Experience Comments in Saudi Dialect Using Deep Learning Techniques

Najla Z. Alhazzani [1], Isra M. Al-Turaiki [2,*] and Sarah A. Alkhodair [1]

[1] Department of Information Technology, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia; najlazalhazzani@gmail.com (N.Z.A.); salkhudair@ksu.edu.sa (S.A.A.)
[2] Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11653, Saudi Arabia
* Correspondence: ialturaiki@ksu.edu.sa

Abstract: Improving the quality of healthcare services is of the utmost importance in healthcare systems. Patient experience is a key aspect that should be gauged and monitored continuously. However, the measurement of such a vital indicator typically cannot be carried out directly, instead being derived from the opinions of patients who usually express their experience in free text. When it comes to patient comments written in the Arabic language, the currently used strategy to classify Arabic comments is totally reliant on human annotation, which is time-consuming and prone to subjectivity and error. Thus, fully using the value of patient feedback in a timely manner is difficult. This paper addresses the problem of classifying patient experience (PX) comments written in Arabic into 25 classes by using deep learning- and BERT-based models. A real-world data set of patient comments is obtained from the Saudi Ministry of Health for this purpose. Features are extracted from the data set, then used to train deep learning-based classifiers—including BiLSTM and BiGRU—for which pre-trained static word embedding and pre-training vector word embeddings are utilized. Furthermore, we utilize several Arabic pre-trained BERT models, in addition to building PX_BERT, a customized BERT model using the PX unlabeled database. From the experimental results for the 28 classifiers built in this study, the best-performing models (based on the F1 score) are found to be PX_BERT and AraBERTv02. To the best of our knowledge, this is the first study to tackle PX comment classification for the Arabic language.

Keywords: text classification; multi-label classification; deep learning; natural language processing; NLP; word embeddings; Arabic; patient experience; PX; LSTM; GRU; BERT

Citation: Alhazzani, N.Z.; Al-Turaiki, I.M.; Alkhodair, S.A. Text Classification of Patient Experience Comments in Saudi Dialect Using Deep Learning Techniques. *Appl. Sci.* **2023**, *13*, 10305. https://doi.org/10.3390/app131810305

Academic Editors: Sang-Woong Lee and O-Joun Lee

Received: 24 August 2023
Revised: 10 September 2023
Accepted: 11 September 2023
Published: 14 September 2023

1. Introduction

In the healthcare sector, measuring the *patient experience* (PX) is a topic of major interest that plays an important role in the safety and well-being of individuals. The Beryl Institute defines PX as "the sum of all interactions, shaped by an organization's culture, that influence patient perceptions across the continuum of care" [1]. Measuring patient experience is important for several reasons. First, it provides valuable feedback on the quality of care provided by healthcare organizations, helping them to identify areas for improvement. Second, it can improve patient satisfaction and engagement, leading to better health outcomes, as well as relieving the burden on medical workers [2]. Finally, it can inform healthcare policy and decision making, ensuring that patient-centric care is prioritized. According to the *Patient Access Leadership Research Report* released by Lumeon in 2021, 90% of survey participants believe that PX is the primary differentiator for hospitals [3].

In the Kingdom of Saudi Arabia, the Ministry of Health (MOH) pays great attention to measuring the patient experience, and a program was launched in 2018 to measure patient experience [4]. The goals of this program include raising the quality of healthcare services, aiding decision makers with insightful reports about patient experience, and developing

standards for patient experience that comply with regional and international standards. Patient experience information is typically gathered using surveys. The survey questions are close-ended with five-point scales, allowing for the measurement of their degree of satisfaction. In addition, there is also an open-ended question that allows the patient to express freely their opinions about their experience during the last visit. This open-ended question is formatted as "please provide additional comments", and the response is mostly free text in Arabic.

Machine learning (ML) algorithms have been widely adopted for text classification in various healthcare fields, including for the analysis of patient experience data [5–9]. However, existing research in the literature has mainly focused on the English language. As such, at present, patient comments written in Arabic are difficult to automatically analyze using traditional machine learning and statistical approaches. Dealing with Arabic text is challenging for many reasons; for example, it has complex grammar due to the presence of different forms of words based on gender, singular, plural, and dual contexts. In addition, there are many dialectal variations, where each has its own morphology, phonology, syntax, and lexical forms [10,11]. Many studies have demonstrated the difficulties associated to processing Arabic text, and some researchers have attributed this fact to the richness and complexity of the Arabic morphology [11,12].

The current approach used by the Saudi MOH is manually categorizing comments based on a pre-defined set of classes. With the huge number of patient comments received from health facilities across the country (i.e., hundreds of thousands), it is difficult to draw insights in a timely manner, thus slowing down the process of addressing patient concerns. Manual classification is a tedious task and humans are subject to making errors; it is also a very time-consuming task and can demand immense effort.

Deep learning (DL) approaches have recently demonstrated success in many fields, such as image recognition, object detection, speech processing, and text classification [13]. Compared to traditional ML, DL models are able to automatically extract complex features from input data. The use of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) has significantly enhanced the state of the art in text classification [14]. Further improvements have been achieved with the emergence of bidirectional encoder representations from transformers (BERT), which is a language representation model based on transformers for pre-training deep bidirectional representations from unlabeled text [15]. BERT has been used in many natural language processing (NLP) use cases, including sentiment analysis [16], text summarization [17], question answering [18], and text classification [19].

In this paper, we address the problem of classifying patient comments written in Arabic using DL- and BERT-based models. A real-world data set obtained from the PX center at the Saudi MOH is utilized for this purpose [20]. We formulate the problem as a multi-label classification task and define it as follows: given a patient comment that is written in Arabic language, it must be classified into one or more of the 25 pre-defined classes corresponding to the Saudi Healthcare Complaints Taxonomy (SHCT) [21].

The contributions of this study are as follows: (1) we investigate the existing solutions for classifying PX data in the Arabic language; (2) we utilize real-world PX data obtained from the MOH of Saudi Arabia; and (3) we develop a total of 28 classifiers using deep learning models, bidirectional long short-term memory (BiLSTM), and bidirectional gated recurrent unit (BiGRU), in addition to BERT-based models, to help in classifying the newly received unannotated comments. One of the challenges faced in this research is the complexity of the used data set, which poses various difficulties in the classification process. The challenges associated with the data set include dealing with dialectal Arabic (DA) and modern standard Arabic (MSA) and the presence of misspelled words, which may pose a challenge in the classification process. Additionally, the PX data set suffers from an imbalanced distribution of labels, which is a very challenging matter to deal with in the context of multi-label classification [22]. Moreover, we intend to classify the comments with no consideration of their sentiment. Figure 1 depicts the challenges associated with

the PX data set. To the best of our knowledge, the classification of PX comments has not previously been covered in the context of the Arabic language, nor Saudi dialects; however, a number of attempts have been conducted in other languages, revealing an acceptable rate of reliability. The results of the classifications will be in favor of reducing the burden on human resources and delivering insights in a timely manner to decision makers.

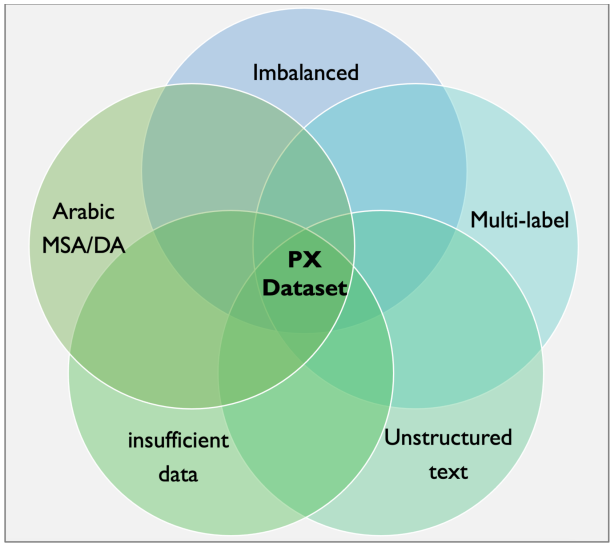

Figure 1. Challenges associated with the PX data set.

The remainder of this paper is organized as follows: In Section 2, we discuss previous works related to text classification. Then, we describe our Arabic patient experience data set and detail the proposed deep learning models for PX classification in Section 3. Our results are presented and discussed in Sections 4 and 5, respectively. Finally, Section 6 concludes this paper.

2. Related Work

In this section, we cover related work in the text classification literature. We investigate the text classification in different domains and in several languages, in order to survey the commonly used techniques. Our discussion of the related literature is structured as shown in Figure 2.

2.1. Text Classification in Healthcare

Text classification in healthcare has received a tremendous amount of attention in the literature. Here, we discuss healthcare-related text classification research focused on languages other than Arabic. Tafti et al. [6] addressed the problem of classifying patient portal English messages using various deep learning models. A data set of 6000 messages was used to train CNN, RNN, long short-term memory (LSTM), and an ensemble model combining the above-mentioned algorithms. The ensemble delivered the best performance of 89.9%, in terms of F1 score. Nawab et al. [7] used a deep learning sequential model to analyze patient experience data in English. The data set of 2830 patient responses to Press Ganey surveys was used to train the model. The model yielded 82% accuracy and an F1 score of 81%. In the pharmaceutical field, Joshi et al. [8] trained ML algorithms to classify drug reviews into ten classes. A data set of 218,000 entries was utilized to train multinomial naïve Bayes (NB), logistic regression (LR), decision tree (DT), extra trees, random forest, and linear support vector classifier (SVC) classifiers. The best results (with 88% precision, 88% recall, and 88% F1 score) was obtained by SVC. In Russian, Alimova et al. [5] used a linear support vector machine (SVM) model to classify the drug side effects into four labels: beneficial effect, adverse effect, symptom, and other. After training on a data set of 5748 drug reviews, the model achieved an F1 score of 73.3%. Khanbahi et al. [9] recently

classified patient experience data obtained from the National Health Service hospitals in England. A data set composed of 69,000 records was used to train various ML algorithms, including decision tree, random forest, SVM, *k*-nearest neighbors (KNNs), NB, and gradient boosted trees. The results indicated that SVM outperformed the others, with accuracy above 62% for each label.

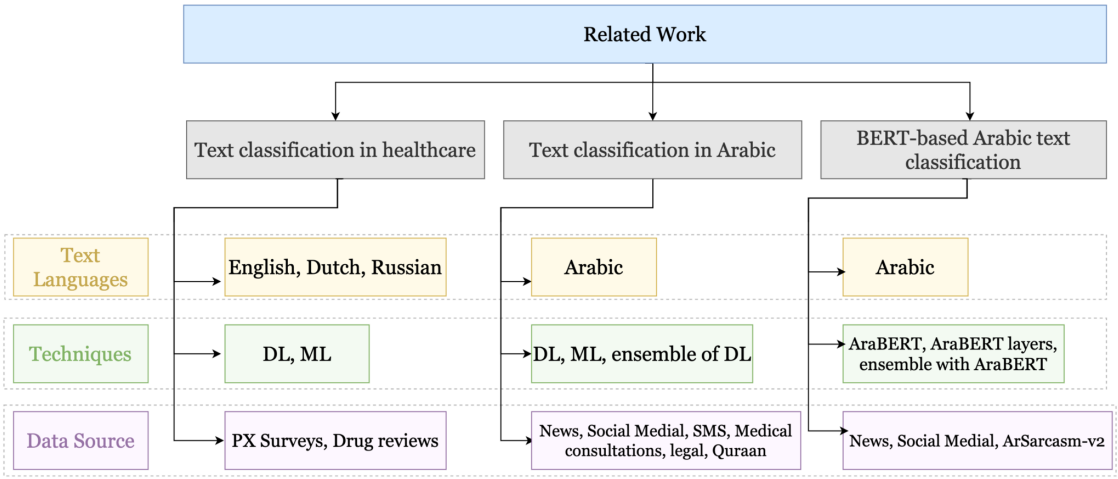

Figure 2. Summary of the related work.

2.2. Text Classification in Arabic

For the Arabic language, many works have focused on text classification. The characteristics of the Arabic language require exploring the relevant approaches considered, as well as comparison of the efficiency for each approach.

El-Rifai et al. [23] developed a binary and multi-label classifiers to classify news data. For binary classification, a data set of 90k articles was used to train several ML algorithms, including LR, KNN, DT, and SVM, where the latter excelled with an F1 score of 97.9%. For the multi-labeled approaches, the data set used comprised 290,000 articles. They adopted shallow-learning techniques as well as neural network techniques, and found that an architecture combining a CNN with a gated recurrent unit (GRU) provided the best result, with an accuracy of 94.85%. Likewise, Alsaleh et al. [24] utilized news data sets and trained their proposed CNN with genetic algorithm (GA) for hyperparameter tuning as a classifier. The authors introduced the Saudi Newspaper Articles data set (SNAD), which contains 45,000 articles in 6 classes. For comparison, the authors utilized the Moroccan News Articles Data set (MNAD) [25], which contains 111,000 articles in 5 classes. The CNN model using the SNAD data set obtained 84% accuracy, while the CNN with the GA optimization resulted in higher accuracy of 88%. The authors also compared their work to a previous work [26] that used the MNAD data set, and found that the result of using global vectors for word representation (GloVe) [27] enhanced the accuracy. On the other hand, Lulu et al. [28] investigated classification of the DA for three dialects of interest: Egyptian, Gulf, and Levantine. The researchers applied deep learning approaches for DA classification and used a subset of the Arabic Online Commentary (AOC) data set [29], which contains 30,000 entries (10,000 for each dialect of interest). The author tested LSTM, CNN, Bidirectional LSTM, and convolutional LSTM models, and found that the LSTM performed best among the others, with 71.4% accuracy. In addition, Wray [30] also focused on DA—specifically, Levantine sub-dialects, which include Palestinian, Jordanian, Lebanese and Syrian. The data set used for their experiment was collected from Twitter and constituted 100,000 tweets, and the author utilized other corpora to extract dialect-specific features. Finally, an SVM model was trained and evaluated with 10-fold cross validation, resulting in 65% accuracy. In the same way, Alorini et al. [10] also concentrated on the DA of the Gulf dialects, in order to conduct binary classification of tweets (as either spam or not).

They used a data set comprised of 2000 tweets from the Gulf region. They considered the use of other features extracted from tweets, such as URLs, hashtags, and a list of profanity words. The authors implemented NB and SVM, where NB delivered better results when combined with the extracted features, obtaining an accuracy of 86%. Additionally, Rachid et al. [11] aimed to classify cyber-bullying within a set of 32,000 deleted comments from an Arabic news website. In their experiment, they utilized AraVec word embeddings [31] with the CNN, LSTM, and GRU combined, which achieved an F1 score of 84%; they also implemented ML approaches and obtained competitive results. On the other hand, Alsukhni et al. [32] utilized a deep learning approach for their proposed models. They employed a multi-layer perceptron (MLP) and an RNN with LSTM to be trained on the Mowjaz data set [33] (of size 9500), which is a multi-labeled data set consisting of 10 classes. The RNN with LSTM surpassed the MLP model, with an F1 score of 82.03%. Furthermore, Ghourabi et al. [34] focused on SMS spam detection, using a deep learning approach that combines a CNN with LSTM and other ML classifiers for benchmarking. The data consisted of 5574 English SMSs and 2730 Arabic SMSs, annotated with two labels as either spam or not, and the researcher developed two types of model. For the machine learning approach, the authors chose term frequency–inverse document frequency (TF-IDF). Alternatively, for the deep learning approach, they used a word embedding model to accommodate the mixed-language data set. A hybrid CNN–LSTM model was then developed, which obtained an accuracy of 98.37%, a precision of 95.39%, a recall of 87.87%, and an F1 score of 91.48%, outperforming the other ML approaches. Al-Laith et al. [35] aimed to review a huge number of tweets with the intention of grasping the public's emotions associated with COVID-19. The authors collected 5,500,000 tweets labeled with six labels for the emotions, along with two classes for the presence of symptoms or not. The researcher utilized word embeddings with an LSTM model, and the symptom classification obtained an accuracy of 75%. Interestingly, Faris et al. [36] proposed a solution to classify medical consultations in Arabic. The authors utilized a data set from the Altibbi telemedicine company, consisting of 1,500,000 consultations, with 75,000 among them labeled and falling under fifteen classes. The authors used many word embedding approaches, and found that developing their own led to the best result. After that, they employed LSTM and BiLSTM models with different variations, in terms of the number of units. The best result was obtained by the BiLSTM with 30 units, which delivered a precision ranging from 83% to 95% for each class. Ikram et al. [37] examined documents in the legal field using a data set of 1452 Arabic documents from the Moroccan Supreme Court for two classes: real estate or traffic. The authors chose four different models for classification of the data: KNN, NB, DT, and SVM with TF-IDF vectorized data. After implementation of the models, they found that SVM delivered the best results, with an accuracy of 98.11% and average F1 score of 98.04%. In addition, Biniz et al. [26] proposed a news classifier using a data set collected from Moroccan news websites, consisting of 111,000 articles falling within five classes. A CNN model was developed, and its hyperparameters were configured to obtain the best output, which resulted in an accuracy of 92.94%. Furthermore, Omar et al. [38] developed a multi-label Arabic text classification model, along with a binary topic classification model, using a Twitter and Facebook data set. The data set, with 44,000 entries in total, includes 4000 entries for each of the eleven classes. Many ML models were developed, among which the linear SVC achieved the best accuracy (of 97.8%) for binary classification, and also achieved the best accuracy (of 81.44%) when combined with TF-IDF in the multi-label experiment. Elnagar et al. [39] also proposed solutions for both binary and multi-labeled classification problems. They created two data sets: the Single-Label Arabic News Articles data set (SANAD), which is annotated with single labels and composed of 110,000 entries; and the multi-label News Articles Data set in Arabic (NADiA), which consists of 450,000 entries and is multi-labeled with 28 labels. For binary classification, the authors implemented nine deep neural network algorithms. The best model among them was the attention-GRU, with an accuracy of 96.94%. For the multi-label classification, attention-GRU also achieved the best accuracy (of 88.68%) on 10 classes. Alhawrat et al. [40] proposed an approach that relies on a multi-

kernel CNN. A total of 15 Arabic data sets with text of different lengths were used. The authors used Arabic Wikipedia fast-text pre-trained word vectors. Then, they trained a multi-kernel CNN model, which achieved an accuracy of 97.58% to 99.90% on the different data sets. Ameur et al. [12] proposed a model that combines the CNN and RNN using the Open-Source Arabic Corpora (OSAC) data set [41], which contains 21,000 documents in 10 classes. Then, they developed many models, and the best result was obtained by the combined RNN-CNN model with static word embeddings, which achieved an F1 score of 98.61%. The authors of [42] studied the multi-classification of Arabic dialect texts from the First Nuanced Arabic Dialect Identification Shared Task (NADI) data set [43], which is labeled with 21 classes representing Arab countries. The researchers developed a classifier that combines LR, NB, and DT using voting with clustering, which yielded the best F1 score of 20.05%. Touati-Hamad et al. [44] aimed to distinguish Arabic Quranic text from non-Quranic text. The data set used for Quran verses was obtained from tanzil.net, while for the non-Quranic text, they used the Arabic learner corpus. The researchers utilized pre-trained word embeddings with a model consisting of hybridized CNN and LSTM layers, which achieved an F1 score of 97.86%, accuracy of 98.33%, precision of 97.86%, and recall of 97.86%.

2.3. Bert-Based Arabic Text Classification

BERT is a contextual word embedding that can be used to derive representations for textual data. This model has been shown to have high performance, and in this section we review some published work that utilizes Arabic pre-trained BERT models. In Ghourabi et al. [45], a classifier was developed to determine the classes for news articles using the Mowjaz data set. The classifier consists of three layers, with an input layer having the size of the hidden layers of AraBERT [46] and an output layer of size 10 (representing the number of labels). Additionally, two other classifiers were developed for comparison: TF-IDF with an SVM classifier and AraVec word embeddings with Bidirectional LSTM. In the testing set, the AraBERT classifier achieved the highest accuracy of 85.1% and outperformed the other classifiers. In their study, Djandji et al. [47] aimed to classify offensive and hate speech Arabic tweets. They utilized a data set of 7839 annotated tweets and conducted pre-processing before fine tuning the pre-trained AraBERT model. The output of AraBERT was then fed to two task-specific dense layers as part of multi-task learning (MTL), which classified whether the tweet was offensive and whether it contained hate speech. The MTL AraBERT model achieved a macro F1 score of 90.15% for the offensive class and 83.1% for hate speech.

In another study, Althabiti et al. [48] aimed to classify tweets as check worthy or not. They used a data set of 4,100,000 tweets to fine tune the AraBERT pre-trained embeddings with 12 BERT layers, followed by a hyperbolic tangent activation function to determine the probability distribution of tokens. The model was tested using different AraBERT versions, and the v0.2-base and v2-base models achieved accuracies of 68% and 69%, respectively. Faraj et al. [49] also employed AraBERT in their research in order to classify whether tweets are sarcastic or not. The authors used an ensemble approach with a hard-voting technique and AraBERT, resulting in an F1 score of 59% for the sarcastic class and an accuracy of 68%. Faris et al. [50] conducted experiments with AraBERT v1 and v2 for multi-class and multi-label text classification of medical consultations obtained from Altibbi, comprising 578,000 consultations with unclear labels. The authors compared the performance of AraBERT + Bidirectional LSTM with Bidirectional LSTM using custom-built static word embeddings (AltibbiVec) and evaluated the performance using precision and recall. The best-performing model was AltibbiVec with BiLSTM, achieving a recall of 54.4%, precision of 26.8%, and F1 score of 35.46%. Uyangodage et al. [51] created a binary classifier to detect harmful social information using an imbalanced data set of over 10,000 tweets about COVID-19 provided by the NLP4IF-2021 [52]. The authors utilized multiple classifiers and found that AraBERT-v2-tokenized achieved the best performance for Arabic, with a macro F1 score of 69.8%.

2.4. Summary

In the literature, the text classification task has been tackled using various approaches. Traditional ML algorithms have been widely investigated for this purpose, and SVM and its variations (e.g., Linear SVC) have demonstrated substantial improvements in performance [5,8,23,30,37,38]. NB classifiers have also been employed for this task [10]. The success of deep learning in many fields has led to the adoption of deep learning algorithms for NLP. CNN—either stand-alone or augmented with other techniques—has shown notably good performance for text classification [6,11,12,24,26,34]. It can be observed that RNNs are not widely used for problems that require sequential dependencies, and many text classification research has used their variations, LSTM/BiLSTM and GRU/BiGRU instead, either stand-alone or in combination with other algorithms [6,11,23,28,32,34–36,39,50]. When an RNN is used, it is often combined with other algorithms [6,12].

The use of BERT for Arabic language text was recently investigated using news and Twitter data sets, and improved text classification results were achieved using AraBERT [45,47–49,51]. Tables 1–3 summarize the related literature focused on binary, multi-class, and multi-label text classification, respectively. Promising results were obtained using DL and AraBERT-based approaches for the classification of Arabic text in many applications; however, to the best of our knowledge, these approaches have not been tested in the context of Arabic PX.

Table 1. Comparison of binary classification models in the reviewed literature. Table legend: A, accuracy; R, recall; P, precision; F, F1 score.

Reference	Language	Best Model	Data Set Source	Classification Type	Result of the Best Model
Tafti 2019 [6]	English	Ensemble (CNN, RNN, LSTM)	Healthcare (PPM)	Binary	F1: 89.9%
Nawab 2020 [7]	English	Deep learning Sequential Model	PX survey	Binary	F1: 81% A: 82%
Alimova 2017 [5]	Russian	Linear SVM	Healthcare (drug reviews)	Binary	F1: 73.3%
El-rifai 2021 [23]	Arabic	SVM	News	Binary	F1: 97.93% A: 97.9%
Alsaleh 2021 [24]	Arabic	CNN with GA	News data sets (SNAD,MNAD)	Binary	SNAD A: 88.71% MNAD; A: 98.42%
Lulu 2018 [28]	Arabic	LSTM	Social Media	Binary	A: 71.4%
Wray 2018 [30]	Arabic	SVM	Social Media	Binary	A: 65%
Alorini 2019 [10]	Arabic	Naïve Bayes	Social Media	Binary	A: 86% F1: 92% P: 81% R: 87%
Rachid 2020 [11]	Arabic	Combination of CNN, LSTM, GRU	Social Media	Binary	F: 84%
Ghourabi 2020 [34]	Arabic	Hybrid CNN–LSTM	SMS text	Binary	A: 98.37% F1: 91.48% P: 95.39% R: 87.87%
Ikram 2019 [37]	Arabic	SVM	Legal text	Binary	A: 98.11% F1: 98.04%
Omar 2021 [38]	Arabic	SVC	Social Media	Binary	A: 97.8% F1: 97.79% R: 97.79%, P: 97.8%
Elnagar 2020 [39]	Arabic	Attention-GRU	News	Binary	A: 95.94%
Alhawrat 2020 [40]	Arabic	Multi-kernel CNN	Miscellaneous		A: (97.58–99.90%)
Ameur 2020 [12]	Arabic	Combined (RNN–CNN)	Online Source Arabic Corpora		F1: 98.61%, P: 98.63% R: 98.58%
Touati-Hamad 2022 [44]	Arabic	Hybrid CNN–LSTM	Quran, Arabic Learner Corpus	Binary	F1: 97.86% A: 98.33% P: 97.86% R: 97.86%
Djandj 2020 [47]	Arabic	AraBERT with MTL	Twitter	Binary	F1: 90.15% (offensive) F1: 83.41% (hate-speech)
Althabit 2021 [48]	Arabic	AraBERT with TanH function	Twitter	Binary	A: 68% (AraBERTv0.2) A: 69% (AraBERTv2)
Faraj 2021 [49]	Arabic	Ensemble (hard-voting) with AraBERT	ArSarcasm-v2 data set	Binary	F1: 59.8% A: 78.3%
Uyangodage 2021 [51]	Arabic	AraBERT	Twitter	Binary	F1: 69.8%

Table 2. Comparison of multi-class classification models in the reviewed literature. Table legend: A, accuracy; R, recall; P, precision; F, F1 score.

Reference	Language	Best Model	Data Set Source	Classification Type	Result of the Best Model
Joshi 2021 [8]	English	Linear SVC	Healthcare (drug reviews)	Multi-class	F1: 88% R: 88% P: 88%
Khanbahi 2022 [9]	English	SVM	Healthcare (PX)	Multi-class	A: 62%+
AL-laith 2021 [35]	Arabic	LSTM	Social Media	Multi-class	A: 75%
Faris 2021 [36]	Arabic	BiLSTM	Healthcare (Altibbi)	Multi-class	A: 87.2% P: (83–95%)
Biniz 2018 [26]	Arabic	CNN	News	Multi-class	A: 92.94%
Aliwy 2020 [42]	Arabic	Ensemble (voting combining LR, NB, DT)	NADI data set	Multi-class	F1: 20.05%
El-rifai 2021 [23]	Arabic	CNN-GRU	News	Multi-label	A: 94.85% F1: 78.86%
Alsukhni 2021 [32]	Arabic	LSTM	News	Multi-label	F1: 83.8%
Omar 2021 [38]	Arabic	Linear SVC	Social Media	Multi-label	A: 81.44%%, F1: 92.0% R: 90.5% P: 93.52%
Elnagar 2020 [39]	Arabic	Attention-GRU	News	Multi-label	Multi: A: 88.86%
Ghourabi 2021 [45]	Arabic	AraBERT	News	Multi-label	A: 85.1% F1: 86.42%
Faris 2022 [50]	Arabic	BiLSTM	Healthcare (Altibbi)	Multi-label	R: 54.4%, P: 26.8%, F1: 35.46%

Table 3. Comparison of multi-label classification models in the reviewed literature. Table legend: A, accuracy; R, recall; P, precision; F, F1 score.

Reference	Language	Best Model	Data Set Source	Classification Type	Result of the Best Model
El-rifai 2021 [23]	Arabic	CNN-GRU	News	Multi-label	A: 94.85% F1: 78.86%
Alsukhni 2021 [32]	Arabic	LSTM	News	Multi-label	F1: 83.8%
Omar 2021 [38]	Arabic	Linear SVC	Social Media	Multi-label	A: 81.44%%, F1: 92.0% R: 90.5% P: 93.52%
Elnagar 2020 [39]	Arabic	Attention-GRU	News	Multi-label	Multi: A: 88.86%
Ghourabi 2021 [45]	Arabic	AraBERT	News	Multi-label	A: 85.1% F1: 86.42%
Faris 2022 [50]	Arabic	BiLSTM	Healthcare (Altibbi)	Multi-label	R: 54.4%, P: 26.8%, F1: 35.46%

3. Proposed Methodology

In this study, we use several DL and BERT-based architectures to build multi-label classifiers for Arabic PX comments. In the following, we describe our data set, pre-processing steps, and the utilized deep learning models, including BiLSTM-, BiGRU-, and BERT-based models. Our proposed methodology is depicted in Figure 3. First, the data set is cleaned, normalized, and then vectorized using static and dynamic word embeddings. Then, several deep learning classifiers are trained using the pre-processed data set, including BiLSTM-, BiGRU-, and BERT-based models. Finally, a testing data set is utilized to evaluate the performance of the classifiers.

3.1. Data Set Description

Our data set was obtained from Patient Experience Center of the Saudi MOH. A total of 21 spreadsheets were provided (file size is 0.59 GB), where each spreadsheet is composed of 1000–4200 manually labeled comments. The records represent responses to the PX surveys—in particular, the answers to the question "do you have any further comments?"—collected from many healthcare facilities across Saudi Arabia. This implies the possibility of them containing different dialects. The entries represent comments collected from primary healthcare centers, inpatient, emergency room (ER), and outpatients. Each record is labeled in terms of sentiment and classification of the comment according to the SHCT [21]. Figure 4 shows the top 100 words present in the data set.

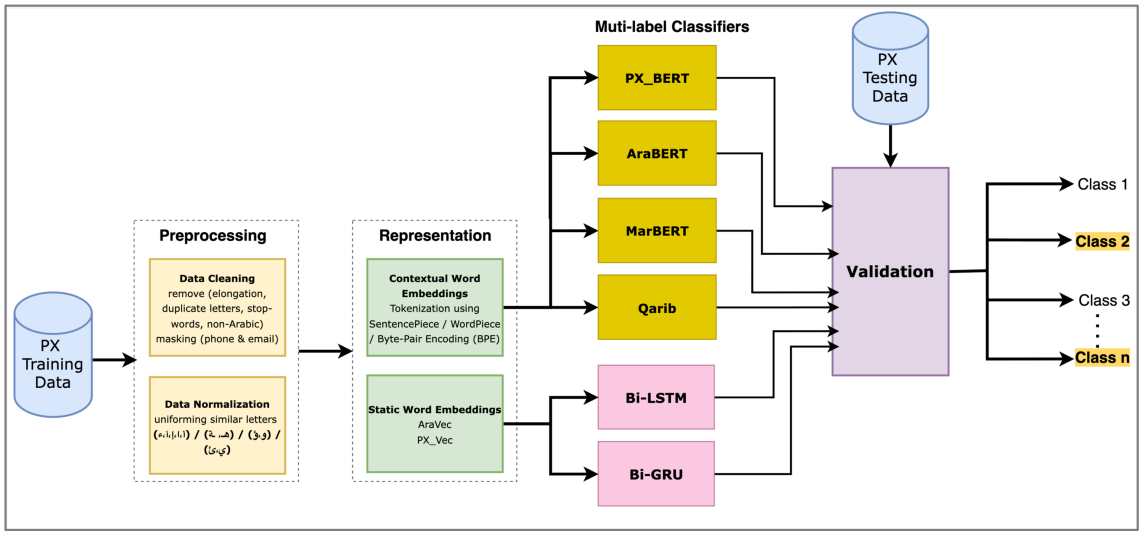

Figure 3. Abstract framework of the proposed methodology.

Figure 4. Top 100 words within the PX 19K all-sentiment data set.

In 2019, the MOH developed the Saudi healthcare complaints taxonomy (SHCT) [21] in order to standardize the classification of patient comments for better analysis. This hierarchical taxonomy provides four levels for classifying patient comments. At the top, the taxonomy begins with the domain, in which there are three main domains: Clinical complaints, relationship complaints, and management complaints. Under each domain comes the second level of the taxonomy, called the category, with seven categories. Then, each category has a number of sub-categories (at this level, there are 25 sub-categories), each of which leads to the final level of the taxonomy, the classification level (consisting of 158 classifications). Table 4 details the number of categories, sub-categories, and classifications for each domain, Table 5 depicts a portion of the SHCT, and Table 6 provides example patient comments.

Table 4. Arrangement of the Saudi healthcare complaints taxonomy.

Domain	Category	Sub-Category	Classification
Clinical	2	8	59
Management	2	11	82
Relationship	2	6	17
Total	6	25	158

Table 5. Sample of the Saudi healthcare complaints taxonomy.

Domain	Category	Subcategory	Classification
Relationships Complaints	Communication	Patient–staff communication	Miscommunication with Patient
			Poor provider–patient communication
			Not involving patient in clinical decisions
			Failure to clarify patient case to their family
		Incorrect Information	Deficient Information
			Communication of wrong information
	Humanness/ Caring	Emotional Support	Inadequate emotional support
			Neglect
		Assault and Harassment	Inappropriate/aggressive behavior
			Provider assaulted patient
			Molesting a patient
			Discrimination
			No apology to the patient

Table 6. Sample patient comments.

Comment in Arabic	Sub-Category	Translation
تعامل كادر التمريض لم يكن على المستوى المطلوب والمأمول بينما الأشياء الأخرى كانت مقبولة جداً	Assault and harassment	The treatment of the nursing staff was not at the desired and expected level, while other things were very acceptable
لبد من مراقبه اصطاف المستشفي ومعرفت اسباب عدم الاهتمام بلمرضي وشكرا لكم	Quality of care	It is necessary to monitor the hospital staff and find out the reasons for not caring for patients. Thank you
اول زياره لي للمركز عندي موعد فحص فايروس كورونا والى الان لي شهر ماطلعت النتيجة اسم رقم الهاتف	Delays	My first visit to the center I have an appointment to check for the Coronavirus and so far I have not seen the results for a month
المبنى ليس مؤهل بأن يطلق عليه مركز صحي	Environment	The building is not qualified to be called a health center

For the purpose of conducting our experiments, we constructed 3 data sets. The first data set contained only records labeled as negative comments, and consisted of 13,000 rows of data and 25 labels. The second data set contained of all the comments, regardless of their sentiment, and comprised 19,000 rows of data with all 25 labels. The third data set contained all of the data regardless of the sentiment and consisted of 19,000 rows of data but with only 20 labels; the least-used six labels were combined into one label that we called 'other'.

In this work, we focus on the classification of the patient comments into the 25 SHCT sub-categories. This choice was based on the need imposed by the PX center, as well as the feasibility of classification on a relatively smaller number of classes. Research has shown that an increase in the number of classes can increase the probability of incorrect classification [23,36]. Table 7 shows the percentage of records for each class in the data set.

Table 7. The percentages of each class within the 19,000 all-sentiment comments data set.

Class	%
Quality_Care	16.43%
Environment	16.17%
Delays	9.34%
Administrative_Policies_procedures	8.07%
Access	5.35%
Medication_Vaccination	5.90%
Examination	6.89%
Staffing	4.38%
Resources	3.76%
Skills_conducts	3.72%
Assault_Harassment	2.44%
PatientStaff_Communication	2.62%
Safety_Incidents	2.37%
Emotional_Support	2.04%
Treatment	1.68%
Patient_Journey	1.7%
Medical_Records	1.28%
Diagnosis	0.81%
Safety_Security	1.35%
Confidentiality	0.72%
Incorrect_Information	1.32%
Referrals	0.57%
Patient_Disposition	0.86%
Finance_Billing	0.30%
Consent	0.07%

3.2. Data Pre-Processing

3.2.1. Data Cleaning

Data cleaning is an important step before model building. It is required in order to remove noise and unnecessary characters, such as symbols, punctuation marks, extra white spaces, English characters, URLs, digits, mobile numbers, emails, new lines, elongations, and stop words. It also protects the privacy of users by removing any personal information. In our data set, cleaning was conducted using regular expressions. The removal of personal information was carried out through multiple steps. The goal was to replace the personal information in patient comments with placeholders. Initially, commonly occurring patterns were identified, including landline numbers, mobile numbers, and email addresses, both in Eastern and Western Arabic numerals. Then, these patterns were used to search for personal information and replace them with the corresponding placeholders, such as $< PHONE >$.

The data set was already labeled using the 158 classifications of the SHCT. There were slight variations in label naming across files and in the same file; for example, the class *Substandard clinical/nursing care* also appears as *substandard clinical-nursing care*. This issue was fixed by transforming annotations into a one-hot encoded format. As we aimed to classify the comments into 25 sub-categories, the 158 classifications were transformed back to their superordinate sub-category as described by the SHCT. Table 8 shows the entries for all sentiments, from which, when cleaned, we found that only 19,000 entries were annotated.

Table 8. Number of all entries, regardless of sentiment.

Patient Journey	Number of Entries
Primary Healthcare Center	9967
Inpatient	8582
ER	7129
Total	**25,678**

3.2.2. Data Normalization

The Arabic language has a rich morphology, and the words in Arabic may have more than a single meaning, and are differentiated with diacritics, or with variations of

the same letter, such as the case of alif and hamza. Data normalization is the process of uniforming text into a more general form, which is performed through unifying the letter Alif آ ، ا ، ء ، إ ، أ to a general form ا. This is also the case for the letter Yaa ى ، ئ ، ي to ي, as well as Haa ـهـ and Taa marbuuta ـة to ـهـ, and Waw ؤ into و. Additionally, the removal of diacritics is a key part of the normalization process. To achieve data normalization, we utilized regular expressions to replace letters with the corresponding general form.

3.2.3. Tokenization

Tokenization is the process of segmenting documents or sentences into smaller segments, which are called *tokens*. Tokenization varies depending on the text language, in order to accommodate for differences in the features of the language. There are several tools that have been specifically built for the Arabic language [53]. There are also more advanced techniques that rely on morphological features for the segmentation of words. Some programming libraries provide morphological tokenizers, such as MADAMIRA [54] and CaMeL Tools [55]. In our case, the type of classifier determines the style of tokenization. For BERT-based models, tokenization is performed specifically for each pre-trained BERT model by replacing links with placeholders, then segmenting the words. Meanwhile, for the deep learning-based models that utilize AraVec, the built-in tokenizers in the *genism* library can be used to load the AraVec model [56].

3.2.4. Data Representation

Feature extraction is the process of transforming text into vectors. There are many approaches to the vectorization of text data.

- **Static word embeddings:** Words are represented using short dense vectors in a multi-dimensional space. This representation is better than traditional techniques, as it can manifest words with similar meaning [57]. In this work, we utilize AraVec, a pre-trained word embedding model published in 2017 by Soliman et al. [31]. It has many variations that were trained on different data sets that compose more than 3,000,000,000 tokens. The used AraVec model was trained with the skip-gram algorithm on the Twitter data set, and has a vector size of length 300 [58]. The skip-gram algorithm was used, as it can grasp the context better than *continuous bag of words* (CBOW) as previously demonstrated in the literature [36].
  Furthermore, we built a patient experience-specific word embedding considering all the comments we obtained, which was 968,985 comments. The same data cleaning and normalization processes described above were followed. Stop words were removed, as they do not add any meaning to the static embeddings. We utilized an existing list of Arabic stop words [59] to eliminate them from the texts. According to previous studies, removing stop words can reduce the size of a corpus by 35–45% while simultaneously increasing the accuracy and effectiveness of text mining applications, thus reducing the overall temporal and spatial complexity of the application [60]. We specified the vector size to be 300, window size 5, and the minimum occurrences of a vocabulary to be 2. We yielded 160,136 unique vocabularies that we utilized in vectorization for some models. It is important to mention that the third quartile for the number of words per comment was 29 for the comments without stop word removal, and 22 for the comments with stop words removed, which implies that the sequence length of the static word embeddings that we built was sufficient for the case at hand. Figure 5 shows the projection of similar word clusters based on the word sense.

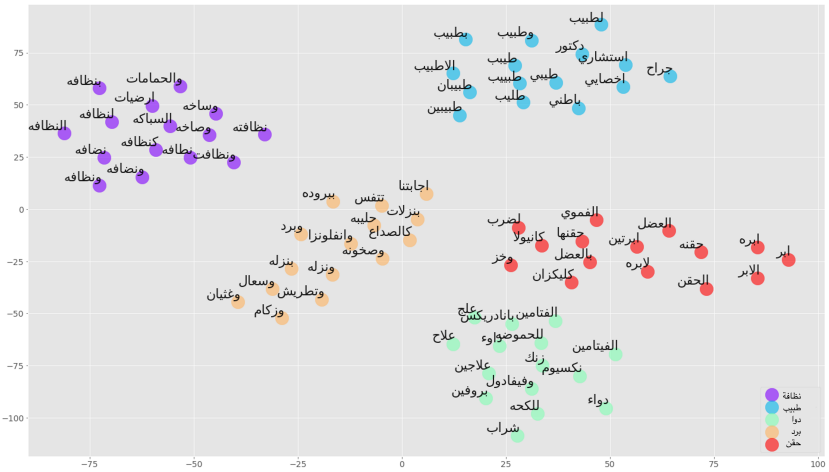

Figure 5. Word embedding similarity clusters from PX_Vec.

- **Contextual Embeddings:** One of the main limitations of static word embeddings is that words with multiple meanings have a single representation. Contextual embeddings solve this problem by capturing the context in which a word appears. Unlabeled data are used to train such models. BERT is a language-dependent pre-trained model that applies the concept of contextual embeddings. It is bidirectional, which means that it captures the context that precedes and follows the represented words [15]. We utilized four contextual embeddings, including AraBERT [46], which is an Arabic implementation of BERT that tokenizes the text using the SentencePiece tokenizer and was trained with 77 GB of the Arabic unlabeled data set; MarBERT [61], which was trained using 128 GB of text and tokenized using WordPiece; and Qarib [62], which is a language model trained using 180,000 tweets and tokenized using byte pair encoding (BPE). All of the above-mentioned models were built using MSA and DA. Table 9 provides the parameters used for pre-training. Moreover, we utilized the unannotated data collected by the PX center of the MOH to pre-train a fourth model: the Arabic PX-specific BERT model (PX_BERT). A total of 968,985 comments written in MSA and DA were utilized. The BERT BPE tokenizer was used for tokenization of the comments. This model was trained with masked language modeling (MLM) head only; we utilized BertForMaskedLM from the transformers Python library, and we configured the model with the values given in Table 10, where the masking percentage was set to 15%, without the next-sentence prediction (NSP) head. This is based on evidence in the literature that no improvement was observed when using NSP in terms of NLP downstream task performance [22].

Table 9. Training configuration used to pre-train the Arabic BERT models.

Model Name	Tokenizer	Approach	Vocabulary Size	Hidden Size	Attention	Hidden Layers	Batch Size	Epochs
AraBERT	SentencePiece	MLM/NSP	64K	768	12	12	512	27
MARBERT	WordPiece	MLM	100K	768	12	12	256	36
Qarib	BPE	MLM	-	768	12	12	-	-

Table 10. Training configuration used to pre-train the PX_BERT.

Model Name	Tokenizer	Approach	Vocabulary Size	Hidden Size	Attention	Hidden Layers	Batch Size	Epochs
PX_BERT	BPE	MLM	50K	768	12	6	32	10

Figure 6 depicts how BERT can be fine tuned for a specific task (in our case, text classification). Our approach to the data representation part is to utilize the pre-trained

AraBERT, MarBERT, and Qarib to reap the benefits of the context-awareness capability that they provide in classifying the PX comments. Then, for comparison purposes, we used static word embeddings to prepare the data for the BiLSTM and BiGRU.

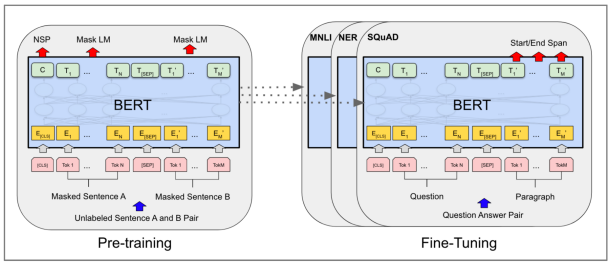

Figure 6. BERT pre-training process, followed by task-specific fine tuning [15].

3.3. Trained Models

In this work, a total of 28 classifiers were trained to classify patient comments. Our classifiers fall into three categories: BiLSTM-, BiGRU-, and BERT-based models. BiLSTM and BiGRU are basically upgraded forms of the recurrent neural network (RNN). The RNN is a feed-forward neural network algorithm that complements deep learning approaches with sequential awareness. It can be described as an algorithm with memory, due to its ability in maintain the previous state of a sequence of a fed vector. The nature of NLP problems requires modeling that has the ability to comprehend a long-term sequence of information. RNN is recurrent, as it feeds the output of its function to the same function as an input, which holds the history of all previous states. Despite having a memory, RNN lacks the ability to maintain information for long-term periods, causing historical information to be forgotten, which means that it cannot be utilized for prediction. This problem is called the *vanishing gradient*.

LSTM and GRU are variations of the RNN that were introduced as solutions to the vanishing gradient problem. Moreover, the two architectures were further enhanced to process the input sequence in both forward and backward directions in order to allow the context to be captured more effectively. BiLSTM and BiGRU demonstrated state-of-the-art performance in many NLP tasks.

3.3.1. Bidirectional Long Short-Term Memory Network

The BiLSTM is a variant of LSTM that takes into consideration the context of the upcoming word sequence. This allows it to capture the most information from the sequence fed into the model. To achieve this, two separate LSTM models are trained, with one of them fed the data in the normal order, and the other in the reverse order [28]. The addition of bidirectionality allows the model to capture longer dependencies when compared to the LSTM [63]. We developed three different variations of the BiLSTM model, with the goal of investigating the effects of hyperparameter tuning and considering multiple factors for best performance:

- **BiLSTM with AraVec static word embeddings:** We utilized AraVec pre-trained static word embeddings, which requires vectorizing the comments into a compatible format of length 300. The vectorized comments were then fed into a Bidirectional LSTM layer through an embedding layer of length 300, using the hyperparameters given in Table 11. The embedding layer of size 300 provides a higher capacity for representing words and allows the model to capture more nuances and semantic relationships in the text. In addition, pre-trained word embeddings such as Word2Vec are often available in 300-dimensional vectors, and using a similar dimension for our embedding layer was expected to facilitate comparison and knowledge transfer. The number of units in the LSTM layer was set to 128, in order to capture more complex patterns and dependencies, as the used data were complex and rich in sequential information. As an optimizer, Adam combines the advantages of both the Adagrad and RMSprop

algorithms. It adapts the learning rate for each parameter, leading to faster convergence and better optimization, making it well-suited to complex models. As the chosen optimization technique was Adam, the learning rate was set to 0.001. Adam automatically adjusts the learning rate during training, and starting with a smaller learning rate (0.001) is generally considered an appropriate choice. Although a lower learning rate can result in slower convergence, it can also lead to a more stable training process. In contrast, higher learning rates may speed up convergence but at the risk of overshooting the optimal solution. A dropout rate of 0.2 is often chosen when training models, representing a moderate regularization level. The batch size was set to 128 for faster training and smoother gradients, and the number of units in a dense layer was set to 25 in order to reduce model complexity, improve training efficiency, and make it less prone to overfitting.

- **BiLSTM with AraVec static word embeddings and hyperparameter tuning:** We aimed to fine tune the hyperparameters to obtain better performance. Many hyperparameter combinations were tested, as detailed in Table 12, using the KerasTuner python library. We implemented 30 grid search trials in order to obtain 30 different combinations of randomly set hyperparameters. Our goal was to determine the hyperparameter combination that leads to the best performance without compromising the time and computing power. Table 13 gives the hyperparameter values for the best model found among the 30 models.
- **BiLSTM with PX-Vec static word embeddings:** PX-Vec word embeddings were built especially for this experiment in order to vectorize the comments. Then, the vectorized comments were fed into a Bidirectional LSTM model through an embedding layer, following the same hyperparameter values mentioned in Table 11, which represent the best hyperparameter set found by the hyperparameter tuning algorithm. Additionally, we carried out cross validation to check the reliability of our model, especially as we used an imbalanced data set.

Table 11. The BiLSTM structure and hyperparameter setup model.

Hyperparameter	Value
Embedding layer	300
Bidirectional LSTM (activation = linear)	128
Dropout	0.2
Bidirectional LSTM (activation = linear)	128
Dropout	0.2
Dense (activation = sigmoid)	25
Optimizer	Adam
Loss	Binary Cross-entropy
Learning Rate	0.001
Epochs	10
Batch Size	128

Table 12. Hyperparameter values for the BiLSTM model experiments.

Hyperparameter	Set of Values
Number of Bidirectional LSTM layers	2, 3, 4, 5, 6, 7, 8
Number of units	8, 16, 32, 64, 128
Activation function	ReLU,tanh,sigmoid,
Recurrent dropout	0.4
Optimizer	Adam, SGD, RMSprop
Dropout	0.2
Loss	Binary Cross-entropy

Table 13. The structure of the best BiLSTM model found through hyperparameter tuning.

Hyperparameter	Set of Values
Embedding layer	300
Bidirectional LSTM (activation = tanh)	32
Bidirectional LSTM (activation = tanh)	32
Bidirectional LSTM (activation = tanh)	32
Bidirectional LSTM (activation = linear)	128
Dropout	0.2
Dense (activation = sigmoid)	25
Optimizer	RMSprop
Recurrent dropout for LSTM	0.4
Loss	Binary Cross-entropy
Learning Rate	0.001
Epochs	10
Batch Size	128

3.3.2. Bidirectional Gated Recurrent Unit

The BiGRU is a variant of the GRU architecture, which is commonly used in NLP tasks, such as text classification and sequence labeling. It consists of two GRU layers, one processing the input sequence forward and the other processing it backward, allowing the network to capture context from both directions. The outputs from both layers are concatenated and fed into a dense layer for the final prediction. Using this deep learning architecture, we developed three different models with the intention of enhancing the performance.

- **BiGRU with AraVec static word embeddings:** Following the same procedure used to build the BiLSTM model, AraVec pre-trained static word embeddings were used to vectorize the comments, which were then fed into a BiGRU model through an embedding layer, using the preset hyperparameter values listed in Table 14.
- **BiGRU with AraVec static word embeddings and hyperparameter tuning**: To obtain the hyperparameter combination that results in the best performance, we experimented with 30 random combinations of the BiGRU hyperparameters listed in Table 15. Table 16 provides the hyperparameter values for the best model found among the 30 trials in this experiment.
- BiGRU with PX-Vec static word embeddings: PX-Vec embeddings were used to represent the comments, then fed into a BiGRU model through an embedding layer. The hyperparameter values mentioned in Table 14 were used. Additionally, cross validation was applied in order to examine the reliability of our model.

Table 14. The BiGRU model structure and hyperparameter setup.

Hyperparameter	Value
Embedding layer	300
Bidirectional GRU (activation = linear)	128
Dropout	0.2
Bidirectional GRU (activation = linear)	25
Dropout	0.2
Dense (activation = sigmoid)	25
Recurrent dropout of GRU	0
Optimizer	Adam
Loss	Binary cross-entropy
Learning Rate	0.001
Epochs	10
Batch Size	128

Table 15. Hyperparameter values for the BiGRU model experiments.

Hyperparameter	Set of Values
Number of Bidirectional GRU layers	2, 3, 4, 5, 6, 7, 8
Number of units	8, 16, 32, 64, 128
Activation function	ReLU, tanh, sigmoid,
Recurrent dropout	0.4
Optimizer	Adam, SGD, RMSprop
Dropout	0.2
Loss	Binary Cross-entropy

Table 16. The structure of the best BiGRU model found through hyperparameter tuning.

Hyperparameter	Set of Values
Embedding layer	300
Bidirectional GRU (activation = tanh)	32
Dropout	0.2
Bidirectional GRU (activation = tanh)	32
Dropout	0.2
Bidirectional GRU (activation = linear)	64
Dropout	0.2
Dense (activation = sigmoid)	64
Recurrent dropout for all GRU	0.4
Optimizer	Adam
Loss	Binary Cross-entropy
Learning Rate	0.001
Epochs	10
Batch Size	128

3.3.3. BERT-Based Model

BERT is a contextual language representation model. It consists of the encoder part from the transformer architecture. The BERT implementation relies on two steps: the first is pre-training, which is the process of training the model on an unlabeled data set to grasp the context of the texts. The second part is fine tuning, which is the process of training the model with labeled data for a specific task [15]—in our case, PX multi-label classification.

To capture language patterns, BERT must be pre-trained on a huge data set of the same language as the labeled data set that will be used to fine tune the model for a specific task. We utilized AraBERT [46], MarBERT [61], and Qarib [62], which are BERT models pre-trained on huge data sets of the Arabic language with both MSA and DA text. This signifies the ability of these contextual models to deal with Arabic text written in both forms. We fine-tuned these models on our pre-processed data sets to investigate the impact of using BERT-based models on the performance measures. Figure 7 depicts the fine-tuning process.

- **Fine-tuned AraBERT:** We fine tuned the AraBERTv02 base model using 80% of the training data and the parameters listed in Table 17. We utilized the AraBERT tokenizer to transform the data into an appropriate format for the fine-tuning process.
- **Fine-tuned MarBERT:** We fine tuned the MarBERT pre-trained model using the parameters listed in Table 17.
- **Fine-tuned Qarib:** We fine tuned the pre-trained Qarib model for the task of multi-label text classification using the parameters listed in Table 17.
- **Fine-tuned PX-BERT:** We built a customized PX_BERT model by pre-training a BERT model using the PX unlabeled data set provided by the PX center at MOH *(PX_BERT pretraining process described in Section 3.2.4).* The pre-trained model was then fine tuned using BertForSequenceClassification from the transformers Python library, with the problem type set to multi-label classification, and the remaining parameters were set as detailed in Table 17.

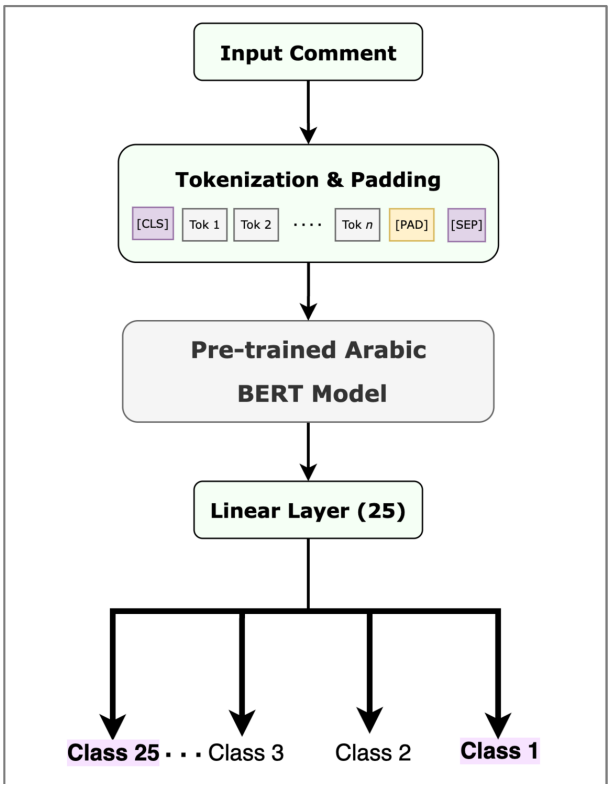

Figure 7. Fine tuning a pre-trained BERT model.

Table 17. Parameters used to fine tune pre-trained BERT models.

Model Name	Version	Batches	Epochs	Learning Rate	Sequence Length
AraBERT	bert-base-arabertv02	8	5	0.00002	512
MARBERT	MarBERTv2	8	5	0.00002	512
Qarib	bert-base-qarib	8	5	0.00002	512
PX_BERT	PX-BERT	16	5	0.00002	512

4. Experimental Results

In this section, we describe the experimental setup, the measures used to evaluate the performance of the developed models, and the obtained results. Subsequently, we discuss our findings.

4.1. Experimental Settings

The developed classifiers were implemented using open-source machine learning libraries, including Keras [64], Tensorflow [65], and Python supplementary libraries. Experiments were carried out on Google Colab Pro Plus, using premium GPU. To simulate a real-life scenario and ensure more robust estimation of the reported results for models and their classification performance, we performed our experiments as follows. We split the data set into a training set (80%) and a testing set (20%). For each classifier, we used the training set to train and build the model. Then, the testing data were used to evaluate the classification performance of the proposed model, in terms of precision, recall, and F1 score. By designing our experiment this way, we ensured that the data set used to validate each model represented new and unseen PX comments. In addition, an external data set of 20 annotated PX comments was requested from the MOH, which we utilized to create predictions using PX BERT, in order to compare the predicted labels against the actual labels. The source code for all of the models is available at https://github.com/NajlaKSA/PX_Classification_2023/ accessed on 1 August 2023.

4.2. Performance Measures

We used four measures to evaluate the classification performance of the 28 multi-label classifiers built in this study: accuracy, precision, recall, and F1 score.

- **Accuracy** measures the percentage of correctly classified observations out of the total number of observations, using the following formula:

$$\text{Accuracy} = \frac{TN + TP}{(TP + FN + FP + TN).}$$

- **Precision** measures the percentage of the correctly classified observations as positives out of the total classified positive observations, using the following formula:

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

- **Recall** measures the percentage of observations classified correctly as positive, out of the total actual positive observations using the following formula:

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

- **F1 Score** is the arithmetic mean over the harmonic mean of precision and recall calculated using the following formula:

$$\text{F1 score} = \frac{1}{n}\sum_{x} F1_x = \frac{1}{n}\sum_{x} \frac{2P_x R_x}{P_x + R_x}.$$

Although we report the four described measures, the F1 score is more relevant to the characteristics of our data set, as it combines both precision and recall.

4.3. Results

We trained 28 classifiers composed of BiLSTM, BiGRU, fine-tuned BERT-based classifiers, and the pre-trained and fine-tuned PX BERT model, in order to classify the PX comments. We present the results of the classifiers depending on the data set used for training in the following.

4.3.1. Negative-Only Data Set (13K)

For the 13K data set (negative-only comments), we trained twelve classification models and reported their classification performance in terms of accuracy, precision, recall, and F1 score as shown in Table 18. The results indicate that all models had relatively low performance, in terms of classification accuracy, ranging between 33.25% for the BiLSTM with PX_Vec embeddings and 10-fold cross validation to 60.24% for the AraBERTv02 model.

In terms of classification precision, the results indicated that BERT-based models excelled over DL models. The results showed that AraBERTv02 outperformed all other models, achieving a classification precision of 81.22%, followed by MARBERT, then Qarib, and finally the PX_BERT model. DL models yielded lower classification performance, in terms of precision, ranging from 40% for the BiGRU with PX_Vec embeddings and 10-fold cross-validation to 12% for the tuned BiGRU with AraVec embeddings.

In terms of recall, all models achieved low classification performance, ranging between 32.54% for the PX_BERT model and 10% for the tuned-BiGRU with AraVec embeddings.

Finally, in terms of the F1 score, the classification performance of the BERT-based models was significantly better than that of the DL models, with a difference of roughly 8%. The PX_BERT model outperformed all other models, with a F1 score of 43.06%, while the tuned-BiGRU with AraVec embeddings yielded the lowest performance, with 10.91%.

Table 18. The performance of models using the negative-only (13K) data set.

Model	Accuracy	Macro F1 Score	Macro Precision	Macro Recall
BiLSTM + AraVec	54.00%	18.49%	19.00%	18.00%
Tuned BiLSTM + AraVec	53.34%	19.00%	19.00%	19.00%
BiGRU + AraVec	47.57%	25.25%	28.00%	23.00%
Tuned BiGRU + AraVec	55.44%	10.91%	12.00%	10.00%
BiLSTM_PX_Vec	54.66%	14.07%	17.00%	12.00%
BiLSTM_PX_Vec (10 folds)	33.25%	15.77%	23.00%	12.00%
BiGRU_PX_Vec	47.84%	27.27%	30.00%	25.00%
BiGRU_PX_Vec (10 folds)	34.78%	30.00%	40.00%	24.00%
AraBERTv02	60.24%	38.83%	81.22%	25.51%
MarBERTv2	56.92%	42.41%	80.51%	28.79%
Qarib	57.60%	41.55%	71.77%	29.24%
PX_BERT	55.14%	43.06%	63.61%	32.54%

4.3.2. All-Sentiment Data Set (19K)

The 19K data set contains all comments regardless of their sentiment. It was used to train all the models, except for the hyperparameter tuned versions for the BiLSTM and BiGRU, due to their immense computational requirements. Table 19 describes the performance of the models using this data set. The accuracy of these models ranged between 53% and 67.97%. Notably, among the DL models, the 10-fold cross-validation versions of BiLSTM and BiGRU yielded lower results, when compared to the models without cross validation (with an approximate decrease of 8% in accuracy). In terms of precision, the obtained values ranged between 38% and 80.62%. It can be seen that all of the fine-tuned BERT-based models achieved better scores, when compared to the DL models. The highest precision value of 80.62% was obtained by MARBERT, followed by PX_BERT and the rest of the fine-tuned BERT-based models. The worst-performing models, in terms of precision, were the DL models, where the lowest precision was obtained by the BiLSTM with PX_Vec embeddings (38%). For the recall, the values ranged between 20% and 37.26%. The best-performing model was Qarib, and the worst was BiLSTM with PX_Vec embeddings. For the F1 score, the values ranged between 26.21% and 47.10%. AraBERT obtained the best F1 score value, while the worst was achieved by BiLSTM with PX_Vec embeddings. In general, the fine-tuned BERT-based models performed better than DL models within this data set.

Table 19. The performance of models using the all-sentiment (19K) data set.

Model	Accuracy	Macro F1 Score	Macro Precision	Macro Recall
BiLSTM_PX_Vec	61.26%	26.21%	38.00%	20.00%
BiLSTM_PX_Vec (10 folds)	53.00%	32.69%	44.00%	26.00%
BiGRU_PX_Vec	62.00%	34.64%	43.00%	29.00%
BiGRU_PX_Vec (10 folds)	54.00%	42.88%	53.00%	36.00%
AraBERTv02	57.95%	47.10%	64.12%	37.22%
MarBERTv2	57.21%	42.68%	80.62%	29.02%
Qarib	67.97%	47.00%	63.65%	37.26%
PX_BERT	55.84%	43.07%	67.45%	31.63%

4.3.3. All Sentiment Data Set (19K with 20 Classes)

In this experiment, we investigated whether reduction in the number of classes would have an effect on the performance of the models. For this purpose, the classes that had fewer than 300 entries were combined into one class, which we labeled as *other*. The combined classes were Consent, Finance_Billing, Referrals, Confidentiality, Diagnosis, and Patient_Disposition. This resulted in 860 entries in the class *other*. The performance of the models when trained using this data set is detailed in Table 20. As can be seen from

the results, the accuracy values for all the models ranged between 40.26% and 60.02%, where the fine-tuned BERT-based models appeared to perform better than the DL models in general. Additionally, the BiLSTM and BiGRU with PX_Vec embeddings were better than the cross-validation versions, exceeding them by at least 12%. In terms of precision, the values ranged between 42% and 66%, with the best being AraBERT and the worst being BiLSTM with PX_Vec embeddings. The BERT-based models mainly performed better than the DL models, where the 10-fold cross-validation versions of the DL model performed roughly 10% better than those without cross validation. However, in terms of recall, the values ranged between 26% and 42%, and the best-performing model is the BiGRU with PX_Vec embeddings with 10-fold cross validation, while the worst is the BiLSTM with PX_Vec embeddings. In terms of F1 score, the scores ranged between 32.12% and 48.7%, the best being AraBERT and the worst being BiLSTM with PX_Vec embeddings. Generally speaking, the performance of all models was improved, when compared to their performance for the 19K data set.

Table 20. The performance of models using the all sentiment (19K, 20 classes) data set.

Model	Accuracy	Macro F1 Score	Macro Precision	Macro Recall
BiLSTM_PX_Vec	54.93%	32.12%	42.00%	26.00%
BiLSTM_PX_Vec (10 folds)	40.71%	42.16%	53.00%	35.00%
BiGRU_PX_Vec	52.67%	40.20%	44.00%	37.00%
BiGRU_PX_Vec (10 folds)	40.26%	47.25%	54.00%	42.00%
AraBERTv02	60.02%	48.70%	66.00%	38.59%
MarBERTv2	57.89%	42.35%	64.33%	31.56%
Qarib	59.10%	46.51%	59.10%	38.34%
PX_BERT	55.74%	45.50%	55.64%	38.49%

4.3.4. Average Results Based on the Various Data Sets

Various data sets were constructed in order to examine the effects of size and number of labels on the performance of the classifiers. For the 13K data set, the averages of all performance measures were lower than those of the other data sets as shown in Table 21. The best-performing model was PX_BERT, in terms of F1 score. For 19K with 25 classes and 19K with 20 classes, the accuracy and precision in the former were slightly higher than in the latter, while the F1 score and recall were higher in the latter. In terms of F1 score, the performance in the 19K data set with 20 classes was better than the 19K data set with 25 classes (by approximately 3%). The F1 score balances precision and recall by calculating their mean, and is a key indicator for the quality of the classification. From our results, we can assume that the increase in data volume and reduction in the number of classes contributed to increasing the F1 score value for all of the models.

Table 21. The averages for all models based on each data set.

Measure	Negative (13K), 25 Classes	All-Sentiments (19K), 25 Classes	All-Sentiments (19K), 20 Classes
Accuracy	50.9%	58.65%	52.67%
Macro F1 Score	27.22%	39.53%	43.10%
Precision	40.43%	56.73%	54.76%
Recall	21.59%	30.77%	35.87%

4.3.5. Computational Time

In addition to evaluating the performance in terms of correct patient comment classification, we measured the computational time aspect of our trained classifiers. The computational complexities of our models on the 13K data set are shown in Table 22. The BiGRU + AraVe model took the least time to train. As expected, with hyperparameter tuning, the training time increased for all models.

Table 22. Computational time (in minutes).

Model	Training Time
BiLSTM + AraVec	48
Tuned BiLSTM + AraVec	1676
BiGRU + AraVec	28
Tuned BiGRU + AraVec	1565
AraBertv02	1102
AraBertv02 (tuned)	1891

5. Discussion

This study involved building 28 classification models to categorize Arabic PX comments, which is a multi-label classification problem. In particular, the classifiers must predict the sub-category or -categories to which each comment may belong. Many attempts to address the text classification problem in various domains (e.g., news, social media, healthcare, and legal) and in different languages (e.g., English, Dutch, Russian, and Arabic) have been detailed in the literature. Some works covered the classification of single labels—either for binary or multi-class classification—while others covered multi-label classification. The problem addressed in this paper is a multi-label classification problem for Arabic PX comments. We highlight these traits to use them as criteria for comparison.

Looking at the multi-label classification research summarized in Table 3, we can see that the majority covered classifying news—which, in nature, tends to be structured text without spelling mistakes—while others covered social media text with a balanced data set, and revealed acceptable scores. At the same time, the most relevant research that covered an imbalanced medical consultation data set revealed comparable performance to our models in terms of F1 score (achieving 35.46%) [50]. We assume that the characteristics of the used data set play a critical role in the quality of the developed classifiers.

Patient comments provide insightful information about patient experience. Such information is rich, compared to the numerical evaluations in patient satisfaction surveys. In our data set, it was observed that the majority of patient comments are related to the quality of care and the healthcare environment. Therefore, it is of vital importance to develop models that can further classify comments up to the fourth level of the Saudi Healthcare Complaints Taxonomy. This is necessary in order to precisely identify patient concerns and address them in an effective and timely manner.

We noticed the lack of PX-related research, especially for the Arabic language. Our work is, to the best of our knowledge, the first to study the problem of classifying Arabic PX comments. The challenges in this field can be attributed to the variability of comment classes, which motivated the MOH to create the Saudi Healthcare Complaint Taxonomy that categorizes comments into 158 classifications under 25 sub-categories. We scoped the problem by classifying comments into these 25 subcategories, which is still a large number compared to that in the reviewed literature, regardless of the variation in domains [23,32,38,39,45,47].

For all data sets, we did not observe any model that had a recall score higher than its precision. This indicates that our models were conservative in their predictions and tended to only predict a positive result when they were confident that they were correct. Among the models trained in this study, it was observed that the fine-tuned BERT-based models performed better than the DL-based models. As for our pre-trained PX_BERT model, the data used for training were relatively small, when compared to other Arabic pre-trained BERT models [46,61,62]. The obtained results demonstrate that increasing the size of the data set used when building language models, such as BERT, can contribute to boosting the performance of the model. This is consistent with evidence presented in the existing literature [46].

Although the data set used for pre-training PX_BERT was small, it can be seen that this model achieved good results on all of the tested data sets. This encourages us to consider reconstructing this model after more data have been obtained. We hypothesize that, for this

particular problem, a domain-specific BERT model may obtain better results, compared to general-purpose BERT models, when created with hundreds of millions of data. There is evidence in the literature that supports the idea that domain-specific BERT models can outperform general-domain BERT models [66].

The data set used in our study exhibited class imbalance, meaning that some classes had significantly fewer examples than others. Handling class imbalance in binary classification has been widely studied [67], and one of the most widely used approaches is SMOTE (synthetic minority oversampling technique). However, handling class imbalance in the multi-label classification context is more challenging and is a less-investigated problem. For a multi-label data set, the minority inputs can have multiple labels that might fall into the minority class as well as the majority class, which makes the over-sampling techniques not feasible as is. According to [22], problem transformation and algorithm adaptation are not effective in handling the data set imbalance problem. In addition, empirical evaluation of re-sampling approaches on six imbalanced multi-label data sets suggested that little improvement can be achieved using the current methods. As a result, addressing class imbalance remains an important challenge in the field of machine learning, and we acknowledge its impact on the results of our study. It is also important to note that this data set was collected during the COVID-19 pandemic, which may have resulted in some patterns that are anomalous with respect to normal conditions and may explain why the majority of comments were labeled with the 'Environment' sub-category, referencing the 'poor cleanliness/sanitization' classification in the SHCT.

Although stop word removal is considered a standard NLP pre-processing step, research on the impact of this step is scarce, particularly the effect of stop word removal on multi-label text classification. A study using English and Portuguese text has shown that there is no significant behavior difference for machine learning algorithms with and without stop word removal [68]. Of course, more studies are required, especially regarding the Arabic language.

As for learning word embeddings, misspelled words and sentences with incorrect grammar in patient comments are treated the same as correct inputs. This limitation has been investigated with regard to the English language [69]; however, more research is required to address this limitation for Arabic grammar.

Other limitations of this work are related to the used data set. First, the sentiment of the comments was not considered. Although this information was provided in the original data set, further validation was required, as the sentiment data were automatically generated. Additionally, the size of the data set should be increased to improve the generalization ability of the model. An important research direction in this context is related to the quantification of the uncertainty of large language models; in particular, in the case of high-risk applications, such models need to avoid frequently overestimating their accuracy when making incorrect predictions.

6. Conclusions

Patient experience is a relatively new concept, which emerged in 2014. Measuring patient experience is crucial for ensuring patient-centered care. It provides valuable insights into the quality of care and enables healthcare organizations to identify areas for improvement. Additionally, it can enhance patient satisfaction and engagement, ultimately leading to better health outcomes.

In this study, several DL and BERT-based models were built to classify PX comments written in Arabic. Our experimental results indicated that AraBERTv02 had the best performance when tested on a data set size of 19,000. We also found that the PX_BERT model performed the best on the 13K negative-only data set and was among the best-performing on the other data sets, with a small margin. These results demonstrate that the domain-specific BERT models are promising and, whenever the data are sufficient, may surpass general-purpose BERT models.

The data set used in this study was collected during the COVID-19 pandemic, which may have resulted in some patterns that deviate from typical circumstances. We also acknowledge that the data set has other limitations, particularly relating to the sentiment of the comments not being explicitly taken into account. Additionally, to enhance the generalization ability of the constructed model, the size of the data set must be increased, and the class imbalance issue must be appropriately dealt with.

As a potential avenue for future studies, investigating the effect of varying token size on the accuracy and results, as well as on the time complexity of the trained models, would be worthwhile. In addition, the use of other language models, such as XLNet, could be studied, which may achieve improved performance in this context. Further hyperparameter tuning could be implemented for the BiLSTM and BiGRU models by performing grid search instead of random search to exhaust all possible hyperparameter combinations. Another approach that could be experimented with is the use of ensemble methods, which could improve the accuracy of the model by combining the predictions obtained by multiple models. These directions for future study have the potential to further improve the performance of the model, thus enhancing its practical utility.

Author Contributions: Conceptualization, I.M.A.-T. and N.Z.A.; methodology, N.Z.A., I.M.A.-T. and S.A.A.; data curation, N.Z.A.; software, N.Z.A.; validation, N.Z.A.; discussion and results, N.Z.A., I.M.A.-T. and S.A.A.; writing—original draft preparation, N.Z.A.; writing—review and editing, N.Z.A., I.M.A.-T. and S.A.A.; supervision, I.M.A.-T. and S.A.A.; funding acquisition, I.M.A.-T. All authors have read and agreed to this version of the manuscript.

Funding: This study was supported by a grant from the Research Center of the Female Scientific and Medical Colleges, Deanship of Scientific Research, King Saud University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the Saudi Ministry of Health for providing the data set used in this research. Special thanks to Dr. Ahmed Sabr, the Director of Development at the Patient Experience Center.

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

References

1. Wolf, J.A.; Niederhauser, V.; Marshburn, D.; LaVela, S.L. Defining Patient Experience. *Patient Exp. J.* **2014**, *1*, 7–19.
2. Ferreira, J.; Patel, P.; Guadagno, E.; Ow, N.; Wray, J.; Emil, S.; Poenaru, D. Patient experience or patient satisfaction? A systematic review of child- and family-reported experience measures in pediatric surgery. *J. Pediatr. Surg.* **2023**, *58*, 862–870. [CrossRef] [PubMed]
3. Lumeon's Report. Available online: https://info.lumeon.com/patient-access-leadership-research-report (accessed on 13 January 2023).
4. Ministry of Health Saudi Arabia. Available online: https://www.moh.gov.sa/en/Pages/Default.aspx (accessed on 3 January 2023).
5. Alimova, I.; Tutubalina, E.; Alferova, J.; Gafiyatullina, G. A Machine Learning Approach to Classification of Drug Reviews in Russian. In Proceedings of the 2017 Ivannikov ISPRAS Open Conference (ISPRAS), Moscow, Russia, 30 November–1 December 2017; IEEE: Moscow, Russia, 2017; pp. 64–69.
6. Tafti, A.P.; Fu, S.; Khurana, A.; Mastorakos, G.M.; Poole, K.G.; Traub, S.J.; Yiannias, J.A.; Liu, H. Artificial intelligence to organize patient portal messages: A journey from an ensemble deep learning text classification to rule-based named entity recognition. In Proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), San Diego, CA, USA, 18–21 November 2019; pp. 1380–1387.
7. Nawab, K.; Ramsey, G.; Schreiber, R. Natural Language Processing to Extract Meaningful Information from Patient Experience Feedback. *Appl. Clin. Inform.* **2020**, *11*, 242–252. [CrossRef] [PubMed]
8. Joshi, S.; Abdelfattah, E. Multi-Class Text Classification Using Machine Learning Models for Online Drug Reviews. In Proceedings of the 2021 IEEE World AI IoT Congress (AIIoT), Virtual, 10–13 May 2021; IEEE: Seattle, WA, USA, 2021; pp. 0262–0267.

9. Khanbhai, M.; Warren, L.; Symons, J.; Flott, K.; Harrison-White, S.; Manton, D.; Darzi, A.; Mayer, E. Using natural language processing to understand, facilitate and maintain continuity in patient experience across transitions of care. *Int. J. Med. Inform.* **2022**, *157*, 104642. [CrossRef] [PubMed]
10. Alorini, D.; Rawat, D.B. Automatic Spam Detection on Gulf Dialectical Arabic Tweets. In Proceedings of the 2019 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 18–21 February 2019; IEEE: Honolulu, HI, USA, 2019; pp. 448–452.
11. Rachid, B.A.; Azza, H.; Ben Ghezala, H.H. Classification of Cyberbullying Text in Arabic. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–7.
12. Ameur, M.S.H.; Belkebir, R.; Guessoum, A. Robust Arabic Text Categorization by Combining Convolutional and Recurrent Neural Networks. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2020**, *19*, 66:1–66:16. [CrossRef]
13. Dong, S.; Wang, P.; Abbas, K. A survey on deep learning and its applications. *Comput. Sci. Rev.* **2021**, *40*, 100379. [CrossRef]
14. Li, Q.; Peng, H.; Li, J.; Xia, C.; Yang, R.; Sun, L.; Yu, P.S.; He, L. A Survey on Text Classification: From Traditional to Deep Learning. *ACM Trans. Intell. Syst. Technol.* **2022**, *13*, 1–41. [CrossRef]
15. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.
16. Wen, Y.; Liang, Y.; Zhu, X. Sentiment analysis of hotel online reviews using the BERT model and ERNIE model—Data from China. *PLoS ONE* **2023**, *18*, e0275382. [CrossRef]
17. Abdel-Salam, S.; Rafea, A. Performance study on extractive text summarization using BERT models. *Information* **2022**, *13*, 67. [CrossRef]
18. Wang, Z.; Ng, P.; Ma, X.; Nallapati, R.; Xiang, B. Multi-passage bert: A globally normalized bert model for open-domain question answering. *arXiv* **2019**, arXiv:1908.08167.
19. Zhang, Y.; Shao, Y.; Zhang, X.; Wan, W.; Li, J.; Sun, J. BERT Based Fake News Detection Model. *Training* **2022**, *1530*, 383.
20. *Patient Experience*; Ministry of Health Saudi Arabia: Ar Riyad, Saudi Arabia. Available online: https://www.moh.gov.sa/en/Ministry/pxmp/Pages/default.aspx (accessed on 15 December 2022) .
21. Saudi Healthcare Complaint Taxonomy. Available online: https://www.moh.gov.sa/en/Ministry/MediaCenter/Publications/Pages/Publications-2019-04-01-001.aspx (accessed on 15 December 2022) .
22. Tarekegn, A.N.; Giacobini, M.; Michalak, K. A review of methods for imbalanced multi-label classification. *Pattern Recognit.* **2021**, *118*, 107965. [CrossRef]
23. El Rifai, H.; Al Qadi, L.; Elnagar, A. Arabic text classification: The need for multi-labeling systems. *Neural Comput. Appl.* **2021**, *34*, 1135–1159. [CrossRef] [PubMed]
24. Alsaleh, D.; Larabi-Marie-Sainte, S. Arabic Text Classification Using Convolutional Neural Network and Genetic Algorithms. *IEEE Access* **2021**, *9*, 91670–91685. [CrossRef]
25. Jbene, M.; Tigani, S.; Saadane, R.; Chehri, A. A Moroccan News Articles Dataset (MNAD) For Arabic Text Categorization. In Proceedings of the 2021 International Conference on Decision Aid Sciences and Application (DASA), Online, 7–8 December 2021; pp. 350–353.
26. Biniz, M.; Boukil, S.; Adnani, F.; Cherrat, L.; Moutaouakkil, A. Arabic Text Classification Using Deep Learning Technics. *Int. J. Grid Distrib. Comput.* **2018**, *11*, 103–114.
27. Pennington, J.; Socher, R.; Manning, C. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1532–1543.
28. Lulu, L.; Elnagar, A. Automatic Arabic Dialect Classification Using Deep Learning Models. *Procedia Comput. Sci.* **2018**, *142*, 262–269. [CrossRef]
29. Zaidan, O.F.; Callison-Burch, C. The Arabic Online Commentary Dataset: An Annotated Dataset of Informal Arabic with High Dialectal Content. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; Association for Computational Linguistics: Portland, OR, USA, 2011; pp. 37–41.
30. Wray, S. Classification of Closely Related Sub-dialects of Arabic Using Support-Vector Machines. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan, 7–12 May 2018; European Language Resources Association (ELRA): Paris, France, 2018; p. 4.
31. Soliman, A.B.; Eissa, K.; El-Beltagy, S.R. AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP. *Procedia Comput. Sci.* **2017**, *117*, 256–265. [CrossRef]
32. alsukhni, B. Multi-Label Arabic Text Classification Based On Deep Learning. In Proceedings of the 2021 12th International Conference on Information and Communication Systems (ICICS), Valencia, Spain, 24–26 May 2021; pp. 475–477.
33. Al-Ayyoub, M.; Selawi, H.; Zaghlol, M.; Al-Natsheh, H.; Suileman, S.; Fadel, A.; Badawi, R.; Morsy, A.; Tuffaha, I.; Aljarrah, M. Mowjaz Multi-Topic Labelling Task. 2021. Available online: https://www.just.edu.jo/icics/icics2021/com/Task%20Description.html (accessed on 15 December 2022)
34. Ghourabi, A.; Mahmood, M.A.; Alzubi, Q.M. A Hybrid CNN-LSTM Model for SMS Spam Detection in Arabic and English Messages. *Future Internet* **2020**, *12*, 156. [CrossRef]

35. Al-Laith, A.; Alenezi, M. Monitoring People's Emotions and Symptoms from Arabic Tweets during the COVID-19 Pandemic. *Information* **2021**, *12*, 86. [CrossRef]
36. Faris, H.; Habib, M.; Faris, M.; Alomari, A.; Castillo, P.A.; Alomari, M. Classification of Arabic healthcare questions based on word embeddings learned from massive consultations: A deep learning approach. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 1811–1827. [CrossRef]
37. Ikram, A.Y.; Chakir, L. Arabic Text Classification in the Legal Domain. In Proceedings of the 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS), Marrakech, Morocco, 28–30 October 2019 ; pp. 1–6.
38. Omar, A.; Mahmoud, T.M.; Mahfouz, A. Multi-label Arabic text classification in Online Social Networks—ScienceDirect. *Inf. Syst.* **2021**, *100*, 101785. [CrossRef]
39. Elnagar, A.; Al-Debsi, R.; Einea, O. Arabic text classification using deep learning models. *Inf. Process. Manag.* **2020**, *57*, 102121. [CrossRef]
40. Alhawarat, M.; Aseeri, A.O. A Superior Arabic Text Categorization Deep Model (SATCDM). *IEEE Access* **2020**, *8*, 24653–24661. [CrossRef]
41. Saad, M.K.; Ashour, W. OSAC: Open source Arabic Corpora. In Proceedings of the 6th International Conference on Electrical and Computer Systems, Lefke, North Cyprus, 25–26 November 2010.
42. Aliwy, A.H.; Taher, H.A.; Abutiheen, Z.A. Arabic Dialects Identification for All Arabic countries. In Proceedings of the Fifth Arabic Natural Language Processing Workshop 2020, Barcelona, Spain, 10 September 2020.
43. Abdul-Mageed, M.; Zhang, C.; Bouamor, H.; Habash, N. NADI 2020: The First Nuanced Arabic Dialect Identification Shared Task. In Proceedings of the Fifth Arabic Natural Language Processing Workshop, Barcelona, Spain, 10 September 2020; Association for Computational Linguistics: Barcelona, Spain, 2020; pp. 97–110.
44. Touati-Hamad, Z.; Ridda Laouar, M.; Bendib, I.; Hakak, S. Arabic Quran Verses Authentication Using Deep Learning and Word Embeddings. *Int. Arab J. Inf. Technol.* **2022**, *19*, 681–688. [CrossRef]
45. Ghourabi, A. A BERT-based system for multi-topic labeling of Arabic content. In Proceedings of the 2021 12th International Conference on Information and Communication Systems (ICICS), Valencia, Spain, 24–26 May 2021; pp. 486–489.
46. Antoun, W.; Baly, F.; Hajj, H. AraBERT: Transformer-based Model for Arabic Language Understanding. *arXiv* **2021**, arXiv:2003.00104.
47. Djandji, M.; Baly, F. Multi-Task Learning using AraBert for Offensive Language Detection. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection. European Language Resource Association (ELRA): Marseille, France, Marseille, France, 11–16 May 2020 ; p. 5.
48. Althabiti, S.; Alsalka, M.; Atwell, E. SCUoL at CheckThat! 2021: An AraBERT Model for Check- Worthiness of Arabic Tweets. In Proceedings of the Working Notes of CLEF 2021—Conference and Labs of the Evaluation Forum, Bucharest, Romania, 21–24 September 2021; p. 5.
49. Faraj, D.; Faraj, D.; Abdullah, M. SarcasmDet at Sarcasm Detection Task 2021 in Arabic using AraBERT Pretrained Model. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, Online, 19 April 2021; Association for Computational Linguistics : Location Kyiv, Ukraine, 2021; p. 6.
50. Faris, H.; Faris, M.; Habib, M.; Alomari, A. Automatic symptoms identification from a massive volume of unstructured medical consultations using deep neural and BERT models. *Heliyon* **2022**, *8*, e09683. [CrossRef]
51. Uyangodage, L.; Ranasinghe, T.; Hettiarachchi, H. Transformers to Fight the COVID-19 Infodemic. *arXiv* **2021**, arXiv:2104.12201.
52. NLP4IF-2021–Fighting the COVID-19 Infodemic. Available online: https://gitlab.com/NLP4IF/nlp4if-2021 (accessed on 15 December 2022) .
53. Farghaly, A.; Shaalan, K. Arabic Natural Language Processing: Challenges and Solutions. *ACM Trans. Asian Lang. Inf. Process.* **2009**, *8*, 14:1–14:22. [CrossRef]
54. Pasha, A.; Al-Badrashiny, M.; Diab, M.; Kholy, A.E.; Eskander, R.; Habash, N.; Pooleery, M.; Rambow, O.; Roth, R.M. MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, 26–31 May 2014; European Language Resources Association (ELRA): Paris, France, 2014; p. 8.
55. Obeid, O.; Zalmout, N.; Khalifa, S.; Taji, D.; Oudah, M.; Alhafni, B.; Inoue, G.; Eryani, F.; Erdmann, A.; Habash, N. CAMeL Tools: An Open Source Python Toolkit for Arabic Natural Language Processing. In Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; European Language Resources Association: Marseille, France, 2020; pp. 7022–7032.
56. Gensim: Topic Modelling for Humans. Available online: https://radimrehurek.com/gensim/index.html (accessed on 8 April 2022).
57. Jurafsky, D.; Martin, J.H. *Speech and Language Processing*; Prentice Hall: Hoboken, NJ, USA, 2000.
58. Soliman, A.B. Bakrianoo/Aravec. 2022. Available online: https://github.com/bakrianoo/aravec (accessed on 3 April 2022).
59. Alrefaie, M.T. Arabic-Stop-Words. 2021. Available online: https://github.com/mohataher/arabic-stop-words (accessed on 1 April 2022).
60. Ladani, D.J.; Desai, N.P. Stopword Identification and Removal Techniques on TC and IR applications: A Survey. In Proceedings of the 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 6–7 March 2020; pp. 466–472.

61. Abdul-Mageed, M.; Elmadany, A.; Nagoudi, E.M.B. 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), Online, 1–6 August 2021; Association for Computational Linguistics : Bangkok, Thailand, 2021; pp. 7088–7105. Available online: https://aclanthology.org/2021.acl-long.0/ (accessed on 3 April 2022).
62. Abdelali, A.; Hassan, S.; Mubarak, H.; Darwish, K.; Samih, Y. Pre-Training BERT on Arabic Tweets: Practical Considerations. *arXiv* **2021**, arXiv:2102.10684.
63. Tian, Z.; Rong, W.; Shi, L.; Liu, J.; Xiong, Z. Attention Aware Bidirectional Gated Recurrent Unit Based Framework for Sentiment Analysis. In *Proceedings of the Knowledge Science, Engineering and Management*; Lecture Notes in Computer Science; Liu, W., Giunchiglia, F., Yang, B., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 67–78.
64. Keras: The Python Deep Learning API. Available online: https://keras.io/ (accessed on 1 January 2022).
65. TensorFlow. Available online: https://www.tensorflow.org/ (accessed on 1 January 2022).
66. Gu, Y.; Tinn, R.; Cheng, H.; Lucas, M.; Usuyama, N.; Liu, X.; Naumann, T.; Gao, J.; Poon, H. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *ACM Trans. Comput. Healthc.* **2022**, *3*, 1–23. [CrossRef]
67. Rezvani, S.; Wang, X. A broad review on class imbalance learning techniques. *Appl. Soft Comput.* **2023**, *143*, 110415. [CrossRef]
68. Gonçalves, T.; Quaresma, P. The impact of nlp techniques in the multilabel text classification problem. In Proceedings of the Intelligent Information Processing and Web Mining: Proceedings of the International IIS: IIPWM '04 Conference, Zakopane, Poland, 17–20 May 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 424–428.
69. Kaneko, M.; Sakaizawa, Y.; Komachi, M. Grammatical Error Detection Using Error- and Grammaticality-Specific Word Embeddings. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Taipei, Taiwan, 27 November–1 December 2017; Asian Federation of Natural Language Processing: Taipei, Taiwan, 2017; pp. 40–48.