Heliyon 8 (2022) e09683

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Automatic symptoms identification from a massive volume of unstructured medical consultations using deep neural and BERT models

Hossam Faris [a,b,c,*], Mohammad Faris [c], Maria Habib [c], Alaa Alomari [c,d]

[a] King Abdullah II School for Information Technology, The University of Jordan, 11942, Jordan
[b] Research Centre for Information and Communications Technologies of the University of Granada (CITIC-UGR), University of Granada, Granada, Spain
[c] Altibbi [1], Amman, Jordan
[d] School of Informatics and Telecommunications Engineering, University of Granada, Granada, Spain

ARTICLE INFO

Keywords:
Deep learning
Multi-classification
Multi-label
Telemedicine
Machine learning

ABSTRACT

Automatic symptom identification plays a crucial role in assisting doctors during the diagnosis process in Telemedicine. In general, physicians spend considerable time on clinical documentation and symptom identification, which is unfeasible due to their full schedule. With text-based consultation services in telemedicine, the identification of symptoms from a user's consultation is a sophisticated process and time-consuming. Moreover, at Altibbi, which is an Arabic telemedicine platform and the context of this work, users consult doctors and describe their conditions in different Arabic dialects which makes the problem more complex and challenging. Therefore, in this work, an advanced deep learning approach is developed consultations with multi-dialects. The approach is formulated as a multi-label multi-class classification using features extracted based on AraBERT and fine-tuned on the bidirectional long short-term memory (BiLSTM) network. The Fine-tuning of BiLSTM relies on features engineered based on different variants of the bidirectional encoder representations from transformers (BERT). Evaluating the models based on precision, recall, and a customized hit rate showed a successful identification of symptoms from Arabic texts with promising accuracy. Hence, this paves the way toward deploying an automated symptom identification model in production at Altibbi which can help general practitioners in telemedicine in providing more efficient and accurate consultations.

1. Introduction

The world has been witnessing the evolution of artificial intelligence (AI) flourishing and influencing widely various real-life domains, such as doctors' clinics. Accurate and timely diagnosis at an early stage of a disease has a significant impact on a patient's life. Moreover, there could be potential errors made by inexperienced physicians when identifying symptoms and determining the diagnosis. This is due to the massive number of symptoms that might be common in different diseases, which in turn confuse physicians. Relatively, this degrades the quality of care and loses patients' trustworthy. Recently, machine and deep learning methods have enriched the diagnosis procedure by the automation of various processes. Such processes are the automatic extraction of symptoms, the automatic detection of diseases, the automatic mapping of symptoms and diagnoses to the international classification of diseases (ICD-10), and others [1, 2, 3, 4]. The ICD-10 is a classification system that uses alphanumeric codes to map diseases into generic categories, including symptoms. This is to promote and standardize the processing, presentation, and transfer of individuals' medical information via healthcare facilities. The ICD-10 codes are periodically refined, the tenth version of the codes is used in Altibbi (ICD-10). The utilization of AI methods in a clinic supporting software promotes the diagnosis process and the early detection of diseases. Computer-aided diagnosis (CAD) systems are emerging computational tools that showed a remarkable ability in assisting doctors during diagnosis. A CAD system is a software-based application that is used as a decision support system for disease diagnosis in clinics. These systems do not just improve the reliability of decisions. But also, reduce the cost of patient monitor-

* Corresponding author at: King Abdullah II School for Information Technology, The University of Jordan, 11942, Jordan. Altibbi, King Hussein Business Park, Amman, Jordan.
E-mail addresses: hossam.faris@ju.edu.jo, inv.hfaris@ugr.es (H. Faris), mohammad.faris@altibbi.com (M. Faris), maria.habib@altibbi.com (M. Habib), alaa.alomari@altibbi.com, omari@correo.ugr.es (A. Alomari).
[1] https://altibbi.com.

https://doi.org/10.1016/j.heliyon.2022.e09683
Received 28 December 2021; Received in revised form 10 April 2022; Accepted 1 June 2022

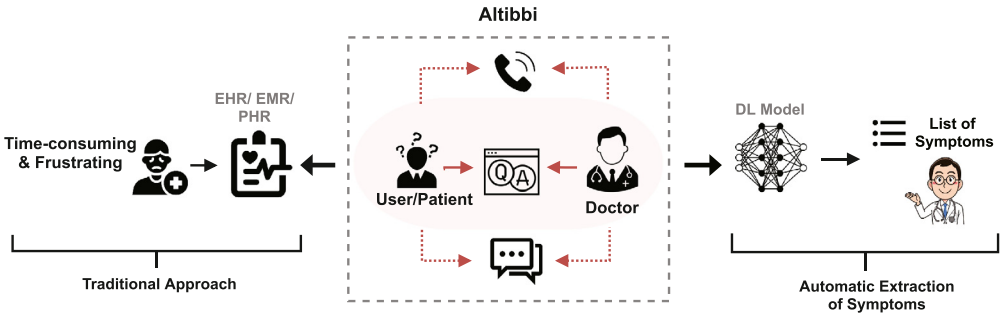

Fig. 1. An illustration of the problem, where users in Altibbi communicate with doctors via phone calls, free question-and-answer, or chat.

ing, save doctors' time, and provide patients with a better quality of care [5].

Our modern society advances in many fields. This correlated with the technological revolution of rising percentages of diverse diseases. Meanwhile, this led to a growing rise of telehealth and telemedicine platforms to help patients and improve their health. An impressive advantage of online healthcare platforms is the large volume of data generated at different scales of variability. Analyzing this rich unstructured data has direct benefits in improving and personalizing diagnosis. Often, users' questions are rich of symptoms as they explain their conditions. Symptoms are signs of an illness or condition a person experiences, such as headache, fatigue, or pain. Essentially, recognizing the symptoms from text correctly results in a higher probability of determining the correct diagnosis. Usually, the patient explains these symptoms with an inherent heterogeneity, ambiguity, unstructured and informal expressions. However, identifying all symptoms is intricate and can be incomplete, and time-consuming, especially for inexperienced physicians [6]. According to Faris et al. in [7], the number of unique symptoms at Altibbi is surpassing 7,000 symptoms, which is huge enough to distract physicians. Furthermore, as Altibbi follows the HL7 (Level 7 Healthcare) standard to exchange clinical data; selecting the appropriate ICD-10 codes is hard and consumes time.

Expressing and recognizing symptoms correctly is a difficult process either for the patient or the doctor [8]. Although, recent research studies have demonstrated the potential of artificial intelligence techniques in disclosing possible symptoms hidden in the text. But, it has also presented open challenges for complex languages, such as Arabic. This paper extends the efforts of previous studies for the extraction of text-based symptoms but in the Arabic language. The Arabic language is the official language of over 300 million individuals across 22 countries scattered in the middle east and north Africa (MENA) region [9]. It has mainly two forms: modern standard Arabic (MSA), and colloquial Arabic. The MSA is used in formal situations, whereas the colloquial is used in usual conversations, where it differs from one country to another and from city to city. Arabic has differences from other languages. It has more alphabet characters and sounds that do not exist in other languages. One of the major challenges when processing Arabic text is the presence of diverse dialects that differ among countries and cities, this means that the same word can be written differently aside from the misspellings, too. This makes Arabic preprocessing a harder process.

This research sheds light on a leading telemedicine platform that is known as "Altibbi[2]". Altibbi provides primary care services for people in the MENA region, where users can chat or call doctors, upload their documents, and surf health-related content in Arabic. Also, they can send their questions as a free question-and-answer service. Intriguingly, Altibbi has around three million consultations stored in their databases. One of the most common forms of raw data in Altibbi is text-based consultations where users ask their questions. Users write their questions in their dialect in free text, however, their explanations can be ambiguous, incomplete, and redundant. Besides, users do not explain thoroughly all encountered symptoms in their questions. When Altibbi's doctors receive such questions, it is followed by a text-based chat or a phone call with the patient. Either after the call or chat, doctors assign possible symptoms and diagnoses. Even though, the process is not smooth as it sounds. First, often, the questions are vague. Patients express their experience with the condition in their informal words where symptoms are not clearly defined. Second, there is a high diversity of possible assigned symptoms among physicians. For the same question, different physicians might assign different symptoms. Even though they might be correct but also they might be incomplete, too. To illustrate, "I have severe pain in the back and leg, and I had an x-ray", this is a sample of a question received at Altibbi. One physician will assign "Back pain" as a symptom, while another might assign "Low back pain". Both are correct, but also missing the "Pain in leg" and "Person consulting for an explanation of investigation findings".

Furthermore, one of the major challenges when dealing with patient medical records is the high percentage of missing data. Often, patients' medical records have a high incompleteness rate of filling in inpatient medical information. This is because doctors have full schedules and short visit times, where there is not enough time for documentation and reporting. In [8], the authors stated that nearly 50% of doctors' time is in data documentation, and Arndt et al. in [6] illustrated that physicians need 4-5 hours a day for interacting with clinical systems and entering data. Although, doctors report the symptoms and diagnoses by selecting the suitable ICD-10 codes which is an essential phase, but also consumes much time. As a result, there is a scarcity of cataloged symptoms on the physicians' side. This points out an immense need to automate the extraction, mapping, and documentation of symptoms with such an intelligent approach.

The large amounts of data in Altibbi databases create a rich resource for AI methods to create intelligent models. Deep learning models based on a massive amount of unstructured data play a viable role in the automatic extraction of symptoms. Accordingly, this paper presents a deep learning model that extracts and suggests a list of symptoms for doctors, as presented in Fig. 1, where the users' questions are the only source of data. The developed methodology formulates the problem as a multi-label and multi-classification problem. For each question, a list of the five most likely symptoms are predicted, which are associated with the highest probabilities.

The model is a variant of recurrent neural networks which is the BiLSTM. The BiLSTM network can preserve the long-term dependencies among words via the use of cell states. Whilst, it holds the relevant information and controls its flow through the neural gates. The BiLSTM network showed efficient performance in handling sequences of data, which located it in an exceptional place for such tasks [10]. Thus and according to Faris et al. in [11], the BiLSTM is used for the identification of symptoms in combination with contextual features (embedding) engineered from the pre-training AraBERT model [12]. The performance of the BiLSTM model based on the AraBERT embedding is compared with different embedding structures of AltibbiVec, which

[2] https://www.altibbi.com/.

is a pre-trained Word2Vec model constructed using Altibbi's data with around three million consultations [13].

The proposed model is evaluated based on precision, recall, and a customized hit rate metric. The constructed model revealed an ability to identify at least one actual symptom out of five suggested for doctors. It is noteworthy to emphasize the objective of symptoms identification in this paper which is to predict the symptoms and not to highlight and markup symptoms as entities in text. Therefore, a list of the most likely predicted symptoms are generated and presented to doctors.

The main contribution of the paper is the proposal of an automatic identification system for medical symptoms of a condition from text-based consultation in the multi-dialect Arabic language. The main purpose of the proposed system is to support medical doctors in providing medical consultations and shorten the time needed in the process. The system is based on the development of a deep learning model based on features extracted from a fine-tuned AraBERT model, and a classification model of the BiLSTM network. This is by the utilization of a massive amount of medical consultations.

The rest of the paper is organized as follows. Section 2 presents recent works in medical information extraction and symptoms extraction. Following this, Section 3 discusses the used material and methods, including the formulation of the problem, the data preparation, the model development, and evaluation. Section 4.2 discusses the results, and Section 5 is a conclusion of the findings.

2. Literature review

Automating the medical symptoms identification process has attracted researchers' attention for years. The continuous advancements in computational techniques including natural language processing (NLP) and deep learning models have paved the way for a much more promising future for a wide spectrum of applications. This section explores and summarizes relevant research studies for the automatic identification of medical symptoms in text, as well as, common algorithms and methods used in the medical NLP domain are manifested. Processing text data is challenging particularly in the medical domain due to a lack of in-domain linguistic resources. However, it is harder when the context is a low-resource language like Arabic. In this regard, research efforts of symptoms identification in written texts will be reviewed and gaps with low-resource languages especially Arabic will be indicated. Further, this review covers the period from 2015 up to the present by searching Scopus and Google Scholar databases.

A fundamental step when building any deep or machine learning model is feature extraction or engineering. When the data is textual, the extraction of features is a bit different where the aim is to create word or text embeddings. Generally in medical NLP and at the feature extraction level, the BERT (as a state-of-the-art model) was used in several studies as in [14, 15, 16, 17], whereas, the Word2Vec model was used in [18, 19, 20]. In contrast, and at the algorithmic level, various deep learning models were widely used in medical NLP. Such as: The BiLSTM/LSTM in [14, 16, 20, 21, 22, 23, 24], the convolutional neural network (CNN) in [19, 25], the capsule network [26], the transformers [27], the ResNet-34 network [28], and the generative adversarial network (GAN) [29]. Medical symptom extraction is a well-known problem in health or medical-related NLP tasks. Various research studies devoted much effort to tackling such a problem, especially in the English context. For example, Jackson et al. [30] followed an NLP approach for the identification of severe mental illness symptoms from discharge summaries. The problem is formulated as classification into five categories (Catatonic, Disorganization, Manic, Positive, and Negative), which is implemented using TextHunter and around 1.2 million electronic health records (EHRs). The authors identified 46 symptoms of severe mental illness with an f1-score of 88%. Also, Du et al. [31] investigated the extraction of symptoms from clinical conversations using a sequence-to-sequence deep learning model. The performance was characterized by an f1-score of (50-80)% depending on the task. In another paper, Eisman et al. [32] studied the identification of Angina symptoms from clinical notes by utilizing pre-trained transformer architectures. For which, the BioBERT and 459 primary care physician notes were used. As a result, the authors recommended the model for the automatic identification of symptoms in clinical decision support systems.

Moreover, Leiter et al. [33] identified symptoms of congestive heart failure based on clinical notes using a deep learning model. In consequence, the identified symptoms were classified into three classes with an f1-score of 71%. Further, Wu et al. [34] extracted symptoms and function profiles of mental disorders based on data documented in EHRs by implementing a dictionary and machine learning approach. Around 500 records were utilized in the model, where it had a satisfactory performance with an f1-score of approximately (75-77)%. In [35], Uddin et al. identified depressive symptoms from text using a deep learning model based on LSTM. The model was trained on two constructed datasets from a public online information channel in Norway, where they were having 11,807 and 21,470 instances. In the developed method, the tokens were represented based on a symptom-based one-hot encoding scheme, where the model achieved an f1-score of 98%. Nonetheless, Wang et al. [36] developed a clinical NLP approach for the identification of symptoms of Coronavirus Disease. The constructed "COVID-19 SignSym" was built based on clinical texts from five databases and a hybrid of deep learning models, curated lexicons, and pattern-based rules. The developed tool showed efficient performance and it is freely available on the Internet.

Roughly, different studies for symptom identification were concerned with general text that is not related directly to the medical or clinical text. For instance, Magge et al. [37] developed a framework called "SEED" for symptom extraction from social media posts. The authors implemented deep learning and transfer learning approach that achieved an f1-score of 85%. Additionally, Yao et al. [38] tried to extract depressive symptoms from an online depression community using a deep classification approach. The objective of the model was to classify the identified symptoms into five categories: emotional, cognitive, motivational, vegetative, physical, and seeking help. For this, network analyses were conducted to find effective features of depression in online communities. Guo et al. [39] presented a general symptoms and diseases inference model using a deep learning approach. The MetaMap is used for the extraction of symptom terms and then they were represented using the term frequency-inverse document frequency (TF-IDF). According to the proposed methodology and regarding the MIMIC-III dataset, the BiLSTM model achieved significant improvement in terms of the area under the curve (AUC) with 85.3% and f1-score with 56.3%. Also, Abulaish et al. [40] constructed a system that is known as "DiseaSE" for symptoms extraction and associations modeling. Eight diseases were considered; which are dengue, malaria, diarrhea, cholera, meningitis, influenza, meningitis, and leishmaniasis. Using TextRank and the PubMed database yielded the identification of new symptoms that were not listed in the Center of Disease Control.

There are other noticeable works in the literature that targeted the problem of symptom identification in other language contexts. For example, in the German context, Schafer et al. [41] proposed a BERT-based model for the identification and extraction of symptoms using German patient monologues. The hybrid approach of BERT and the Curriculum Learning and Augmented Descriptions method achieved the highest f1-score of 90%. Also, in the Italian context, Polignano et al. [42] proposed a health bot that processes symptoms, and suggests diagnoses and treatments. Moreover, Wada et al. [43] attempted to extract symptom names from general medical text using a deep learning approach in Japanese. However, as the training texts were curated from the general web, the identified symptoms have no standard medical terminologies. Faviez et al. [44] detected symptoms in tweets using a fuzzy matching approach in French. Extracting coronavirus symptoms was presented as an application with a precision of detection of 81%.

Prior studies have demonstrated the potential of automatic symptom extraction in different languages. However, to the best of our knowl-

edge, there are very few attempts were dedicated to the analysis and identification of symptoms embedded in text in the multi-dialect Arabic context. For instance, Alghamidi et al. [45] proposed a model for the prediction of depression symptoms in Arabic psychological forums. Different models were experimented with which depended on machine learning or lexicon-based and rule-based approaches with different embedding algorithms. The models showed efficient detection of posts with depression symptoms of a recall of 80%, and precision of 79%. Also, Alotaibi et al. [46] created a big data model for the detection of symptoms and diseases from Twitter posts in the Arabic context in Saudi Arabia. Around 18.9 million tweets were used to build a machine learning model, which achieved a performance higher than 80% in terms of accuracy and f1-score.

In consequence, this research is proposed to identify the symptoms from users' questions which are written in multi-dialect Arabic using a deep learning model. The next section demonstrates the used methods and datasets.

3. Methodology

This section discusses the methodology that is followed in this work to develop the proposed symptoms identification system. The methodology consists of five main stages: problem formulation, data collection and preparation, feature engineering, model development, and evaluation. Fig. 2 presents an abstract overview of the proposed system. In the following subsections, the five stages are discussed.

3.1. Problem formulation

This paper presents the problem of symptom identification that is formulated as supervised multi-label multi-classification. In supervised multi-label classification, multiple labels are assigned to a sample ($x \in X$), where the labels are in the set (L). Samples are presented as numerical vectors of features, hence, it is a mapping function from the feature space X to L; such that $h : X \rightarrow P(L)$, where P is the probability.

Multi-label classification can be handled broadly by two different techniques; transformation methods, and adaptation methods. Transformation methods treat each label as a binary classification problem with either one-versus-all or one-versus-one strategies, where outputs from all classifiers are combined to produce the final set of labels for a test example. A critical drawback of transformation methods is that they do not consider the relationships between labels and treat each of them independently. Also, transformation methods require building a higher number of classification models on smaller sets of data. Adaptation methods build on the existing traditional (single-target) algorithms to directly handle the associations among labels. For example, adapting the neural network to use a new error function, or using a new splitting criterion in tree algorithms [47]. As adaptation methods can better resolve dependencies among labels, various neural network-based approaches constructed in the literature have shown successful ability in handling the multi-label classification problem [48, 49].

Generally, in neural-based adaptation methods, the output can be extracted by two different implementations. First, it is from one output dense layer where the number of neurons is the number of labels. Second, it is from multiple output dense layers where each layer corresponds to a label. In both implementations, the activation method is set to Sigmoid which outputs a probability for each label independently in the interval [0, 1] as in Equation (1). All generated probabilities are not constrained to have a sum of one, for example, given that the final output layer has four neurons for four labels (e.g., symptoms) with raw outputs of [-1.5, 0.2, -0.7, 1.3]. After Sigmoid activation, the output probabilities are 0.182, 0.549, 0.332, and 0.786 of the four symptoms, respectively.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{1}$$

Table 1. Summary of the used datasets.

The Dataset for Embedding	
Number of questions	3,310,996
Vocabulary size before preprocessing	1,032,093
Vocabulary size	171,385
The Dataset for the Classifier	
Data size after removing duplicates	578,941
Data size after removing infrequent labels	567,399
Data size after sampling 5000 row per label	501,004
Number of unique symptoms before preprocessing	4,689
Number of unique symptoms after preprocessing	2,348
Vocabulary size	165,781

In this paper, the problem is formulated by an adaptation approach using a deep recurrent neural network with one output dense layer and Sigmoid activation. The choice of adaptation methods is that the model will train on the entire dataset without the need for manipulating or transforming it, while also considering the associations among symptoms, and better suggesting relevant symptoms together. Also, handling the multi-labeling by a single output dense layer instead of a multiple is envisaged to reduce the complexity of the model which better suits a light model's production.

3.2. Data collection and preparation

All used data in this work was drawn from Altibbi's databases. Two datasets are used; one for building the embeddings, and the other is for building the classifier of the symptom identification system. For building the embeddings, the collected data is all from Altibbi content and accounts for 3,310,996 questions. Whereas, for building the classifier, 578,941 consultations are used. The medical consultations are questions asked by patients from different Arab countries; such as Jordan, Egypt, and Saudi Arabia. The consultations are in the form of questions in various Arabic dialects. For example, such consultation is "طفلتي عمر ها شهور ابغي يفتح شهيتها وشنو اوكلها وزنها كيلو", which is translated into "My baby is 6 months old, I want to increase her appetite and what type of food to give her, she weighs a kilo".

The two datasets are preprocessed by applying different preprocessing steps. This includes removing duplicate questions and cleaning the text from punctuations, symbols, English characters, and numbers. Also, removing the elongation of characters and the diacritics. Furthermore, the text was preprocessed at the label level. As questions are accompanied by a list of symptoms; questions with infrequent symptoms were removed. The infrequent symptoms are the ones that appeared in less than ten questions. The number of initial symptoms in the dataset which were used for building the classifier was 4,689, while after preprocessing, the unique symptoms in 501,004 consultations were 2,348. Fig. 3 shows the count of the most frequent 25 symptoms across this dataset.

A statistical analysis at the question-level is visualized in Fig. 4. It is clear that the longest question has 175 tokens, where most of the questions are not higher than 30 tokens. In subfigure (b), most of the questions are accompanied by only one symptom, given that the average of symptoms across the dataset that is used for building the classifier is about 1.5 symptoms per consultation.

The following Table 1 summarizes the main attributes of the datasets.

3.3. Feature engineering

For feature engineering, two approaches we developed, evaluated and compared. The first is to extract contextual embeddings from AraBERT by fine-tuning them. The fine-tuning of AraBERT is performed by freezing the entire architecture and adding a new final classifier layer (the BiLSTM) where only its parameters will be updated using a dataset of approximately half a million consultations. The second is to train the Word2Vec (AltibbiVec) model from scratch using around three million Arabic medical consultations and use the extracted embeddings with the

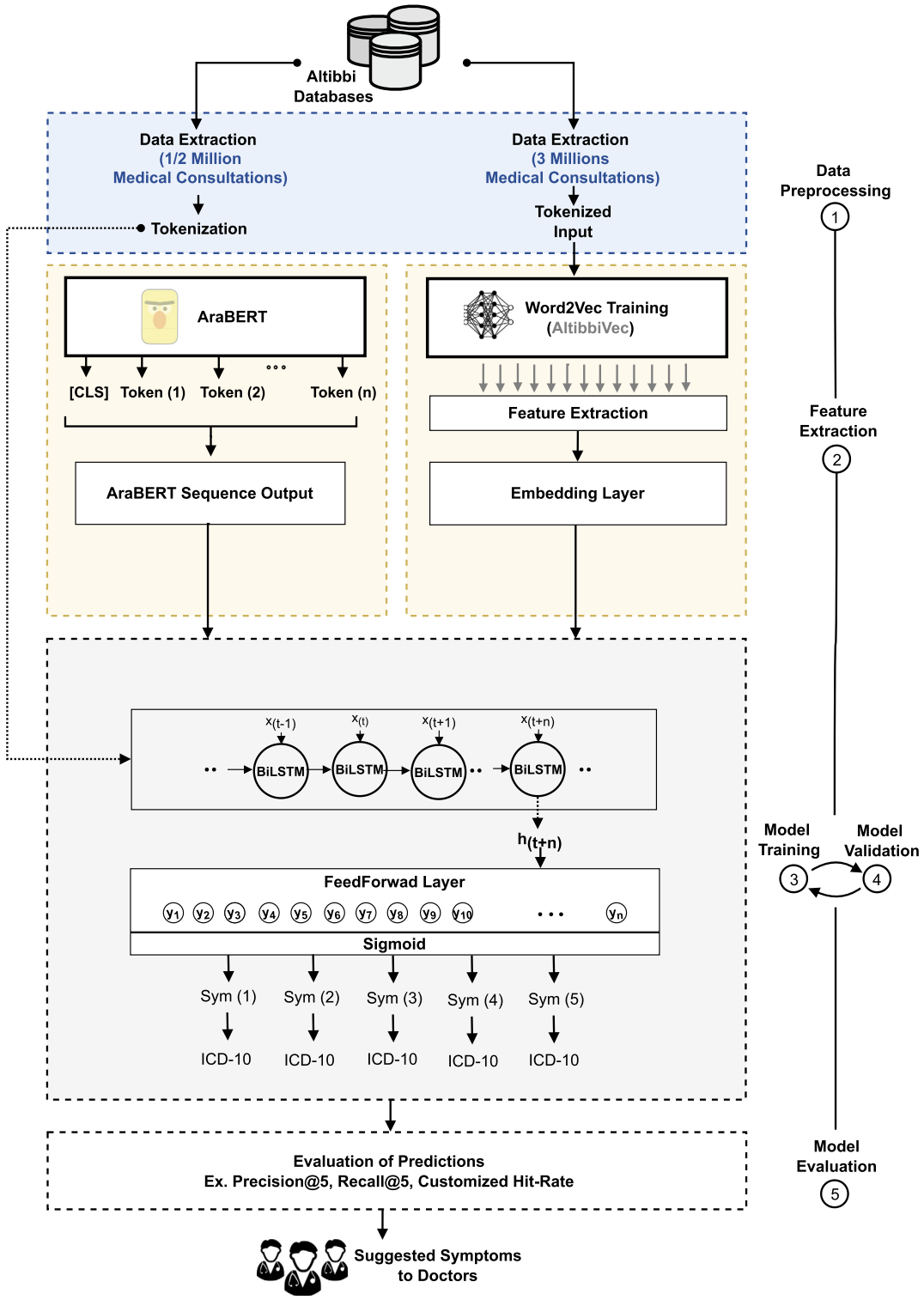

Fig. 2. An illustrative description of the proposed intelligent symptom identifier.

classification dataset to train and evaluate the BiLSTM. This subsection explains and provides an overview of features engineering approaches including AraBERT and AltibbiVec models.

3.3.1. AraBERT embedding

BERT as a language representation model learns deep bidirectional representations of text by jointly considering the left and right contexts. The structure of BERT is deep Transformers designed to pre-train on a huge amount of unlabeled text and learn by self-supervision when fine-tuning with labeled data. A Transformer network is a composition of two sub-networks; the encoder, and the decoder. The encoder is responsible for creating a vector representation of the input. Whereas, the decoder takes the encoder's output as input to generate the target output. BERT was developed by Google in 2018 [49] and used the encoder part only of the Transformer in two configurations; the base and large. Essentially, the embedding of tokens can be extracted from the last layer, any of the encoder layers, concatenating or summing all layers, or even considering only the last four layers as in [49].

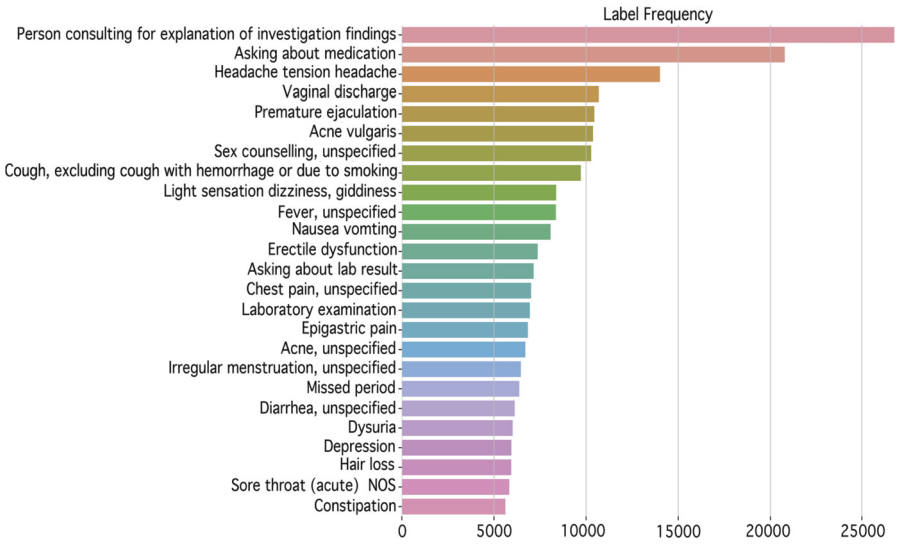

Fig. 3. The frequency of the most recurring symptoms across the dataset.

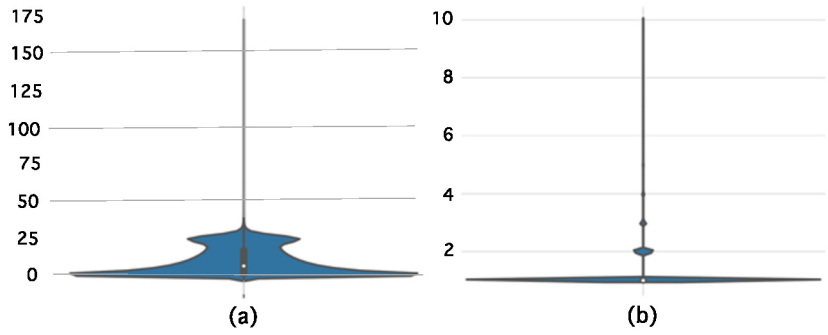

Fig. 4. The distribution of questions lengths (a) and the number of symptoms (b).

AraBERT is a pre-trained BERT architecture in the Arabic context [12]. It was pre-trained on around 12 million new articles in Arabic and was evaluated on three different tasks; sentiment analysis, question-answering, and named-entity recognition. In this paper, two versions of AraBERT were used; the AraBERT-base-v1 and AraBERT-base-v2. The main difference between the two versions is the size of the data. For the first version, the size of the data in terms of sentences is 77 million. It has 2.7 billion words and the model's size is 23 GB. Whereas, the second version, is 200 million sentences, 8.6 billion words, and 77 GB in size. Furthermore, the first version is trained across 1.2 a million steps and the second version is over 3 million steps. Meanwhile, both of them have 136 million parameters and were trained using a tensor processing unit (TPU). The two versions were fine-tuned by freezing all the entire encoding layers and updating only the weights of the final classification layer. The sequence output from the last encoder layer in AraBERT is drawn and fed into the BiLSTM layer, where this sequence contains the embedding of all its tokens.

3.3.2. AltibbiVec

Embedding models convert text-based data into numerical vectors; these numeric representations of words encompass statistic and semantic relationships, where similar words which appear in the same context should have similar vector representations, and hence, a high similarity score. The Word2Vec model is a neural-based embedding model that is capable of capturing the the semantics of words. The Word2Vec model has two implementation structures: continuous bag-of-words (CBOW), and skip-gram (SG) [50], which both are used and implemented in this work.

AltibbiVec embedding model stands on the Word2Vec architecture and was used and compared with the AraBERT embedding. AltibbiVec is a Word2Vec embedding model that was trained from scratch based on Altibbi's medical consultations. The model was trained using around three million medical questions with 1,032,093 unique words. Also, six structures of the model were implemented with the CBOW and SG structures, at three embedding dimensions: 50, 100, and 200. AltibbiVec is used to create an embedding matrix of the tokens embedding which is then used to form the embedding layer.

3.4. Classification model development

The proposed symptoms identification system mainly consists of first the constructed embeddings, and second the classifier (BiLSTM). The extracted embeddings as mentioned in the previous subsections are used as an embedding layer and fed into the BiLSTM.

The BiLSTM has a chain structure of units, where each unit is composed of a memory cell, input gate, output gate, and forget gate. Generally, the structure of the gate includes of element-wise multiplication operator and a Sigmoid function. If the Sigmoid's output is zero means the information will be discarded, while one means the data flows. Given a time step t, the BiLSTM network takes as an input x_t and generates at the end the hidden state h_t that is through a Softmax or Sigmoid layer produces the output y_t. The chain structure of the BiLSTM/LSTM units and their gates can process long sequences of data and preserve relevant information by the cell state. The purpose of the cell state is to hold the information in the cell while regulating and controlling the flow of information through the gates. This ensures moving significant and relevant information through the cell and discards others. The BiL-

STM network is constructed from forward and backward connections, which means it utilizes the previous (historical) and next (future) information in the hidden states. This makes it a suitable algorithm in this context.

In order to utilize the AraBERT embeddings, the tokens embeddings of each corresponding sequence are extracted from the last layer in AraBERT. Then, the embedding of the first token in the sequence which is the "[CLS]" is extracted and fed into the BiLSTM. The embedding of the "[CLS]" token is a contextual embedding that encodes the whole sequence. In contrast, at AltibbiVec, the learned embeddings of tokens are used to create an embedding matrix which is used to build an embedding layer for the BiLSTM. In this case of Word2Vec, the questions were tokenized using the Keras library [51], and the maximum sequence length was set to 50. The final classification model encompasses the embeddings from either AraBERT or AltibbiVec, the BiLSTM network, and the final output dense layer which has a number of neurons equals the number of labels. This classification model is trained and the parameters updated using the smaller dataset that is intended to build the BiLSTM classifier. A 10% of this dataset is used to evaluate the performance of the model in predicting the symptoms. This is demonstrated in the following subsection.

3.5. Model evaluation

The symptoms identification model is evaluated based on two types of evaluation measures, statistical-based, including precision@k, recall@k, and a customized hit rate. As well as by depending on a subject matter expert who is a verified and certified medical coder. As the trained model identifies symptoms, the objective is to predict the most likely top-k symptoms and present them to doctors. The precision presents the ratio of how many predicted symptoms are actually in the ground truth, as presented in Equation (2).

$$Precision = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap h(x_i)|}{|h(x_i)|} \tag{2}$$

The recall is the ratio of how many actual symptoms were predicted (as in Equation (3)). Given that the number of examples in the testing dataset is n, Y_i is the ground truth label, $h(x_i)$ is the predicted value.

$$Recall = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap h(x_i)|}{|Y_i|} \tag{3}$$

The actual labels are the verified unique symptoms. However, the measures of precision and recall are calculated in the first top k symptoms instead of all of the symptoms. Besides, regarding the operations team at Altibbi, doctors prefer to see a maximum of five suggested symptoms to select from them. Therefore, the precision and recall are calculated at values of k from 1 to 5. To illustrate, given a consultation that is labeled by 3 actual symptoms and a suggested list of symptoms presented to doctors is of length 5. If only two of the five suggested symptoms are correct, then the precision@5 is (2/5) and the recall@5 is (2/3). In this regard, the model is evaluated by computing the precision@1, precision@2, precision@3, precision@4, and precision@5. Also, similarly is for the recall.

Moreover, as the number of symptoms in the ground truth across consultations varies, the testing dataset was divided into groups. So, Group (1) is labeled only by one correct symptom, Group (2) by two symptoms, and likewise up to Group (5) with five symptoms. The number of consultations in the Groups from (1) to (5) is 44,938, 4,127, 805, 176, and 35, respectively. Subsequently, the model was evaluated based on a customized hit rate metric. The calculation of the customized hit rate is based on groups of data G and the probability of prediction P. Hence, the model predicts the likelihood of predicting correctly all the expected symptoms given the number of actual symptoms. For example, at G2, the actual expected symptoms are two, the probabilities of predicting at least one of them and two of them correctly are computed. In this regard, for (G1, P1); the hit rate is predicting the one actual symptom correctly averaged over all the samples in G1. Also, for (G4, P3) is the rate of predicting at least three symptoms correctly over all samples in G4.

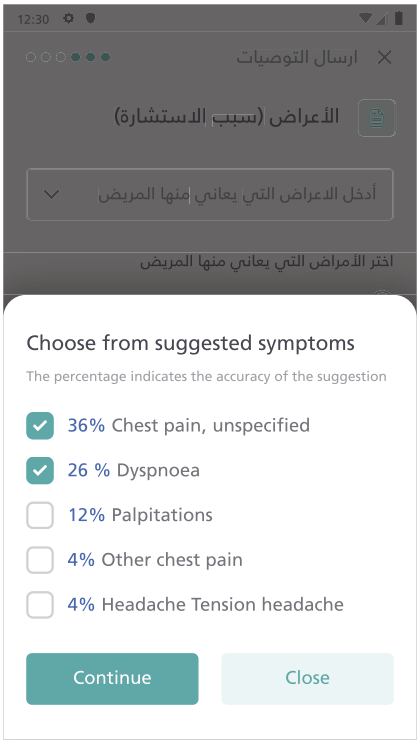

Fig. 5. A demonstration of deploying the symptoms identification model to production in Altibbi.

In addition, the model was evaluated manually by expert doctors who compared the model's suggested symptoms with the symptoms assigned by general practitioners for a sample of consultations. The model is deployed on Altibbi's mobile app for general practitioners to use. A screenshot of the deployed model is shown in Fig. 5. After finishing the chat or the call with the patient, the doctor fills a recommendation with symptoms, diagnoses, required labs, and medications. During this, suggested symptoms from the model are displayed for general practitioners to select from. A sample of 500 consultations was provided to expert doctors for evaluation. The consultations have the actual correct symptoms assigned by general practitioners and the suggested list of symptoms from the model. The role of experts was to check if any of the suggested symptoms were correct and met the actual symptoms labeled by the general practitioner.

4. Experiments and results

This section presents two main experiments at the feature engineering level; the first is to extract contextual features from the AraBERT model and then fine-tune a BiLSTM classification layer. The second is to use AltibbiVec embedding model for feature engineering and then to fine-tune the BilSTM network, too. The implemented experiments were evaluated using precision, recall, and a customized hit rate. The following subsections will present the configuration settings to implement the experiments and a discussion of the results.

4.1. Experimental setup

The structure of the model consists of an embedding layer, a BiLSTM layer, and a dense output layer. Different embedding models were experimented with and compared to AraBERT. Primarily, this includes the

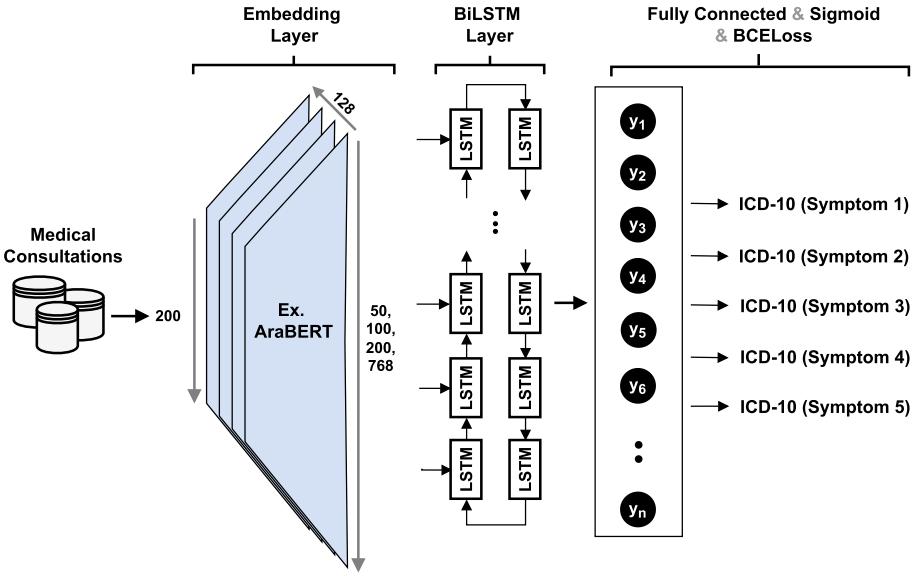

Fig. 6. A description of the designed methodology. In which, different dimensions of the embedding models are used, i.e., 768 for AraBERT and (50, 100, 200) for AltibbiVec.

Table 2. Parameters settings for fine-tuning AraBERT. **Keys:** (L.R.) is the learning rate, (B.S.) is the batch size, (M.T.C) is the maximum token count, and (Dim.) is the embedding dimension.

Parameters	Value
Optimizer	Adam
L.R.	0.001
Activation	Sigmoid
Epochs	10
B.S.	128
Loss	BCELoss
M.T.C	200
Dim.	768

Table 3. Parameters settings for AltibbiVec.

Parameters	Value
Window size	40
Minimum count frequency	5
Down sampling	0.01
Epochs	30

original two versions of AraBERT, and other versions of AltibbiVec at CBOW and SG structures at three different dimensions (i.e., 50, 100, and 200). Table 2 shows the parameters' settings for fine-tuning AraBERT. Initially, AraBERT embeddings were extracted from the base version of the model and the last encoder layer. The input of the embedding layer is a tensor of first the batch size, the sequence length, and the embedding dimension.

Regarding AltibbiVec, Table 3 recaps the attributes used in training AltibbiVec. The words with a frequency less than five were dropped, and the context of words was captured at a window size of 40 according to Silberztein et al. [52]. Furthermore, the number of training epochs was set to 30.

For BiLSTM, the optimizer is set to Adam, the number of BiLSTM layers is one, the activation function is Sigmoid, the number of epochs is 50, the batch size is 128, and the loss function is the Binary_Crossentropy. Additionally, as the number of LSTM units influences the performance, the BiLSTM is tuned based on ten different numbers of units as presented in Table 4. For which, the evaluation is based on a customized hit rate metric. It is clear from the table that the hit rate increases slightly as the units increase up to 40 where then it relatively remains

Table 4. The results of fine-tuning BiLSTM based on the number of units.

No. of Units	Recall
8	0.442
16	0.496
24	0.514
32	0.518
40	0.523
48	0.523
56	**0.529**
64	0.525
72	**0.529**
80	0.522

constant at approximately 52%. Moreover, when the units were 56 or 72, the model had an equal and maximum hit rate of 52.9%. The hit rate is computed and will be presented in the next subsection on different precision levels. Since increasing the number of units also increases the model's complexity, so the number of units was set to a minimum of 56.

The neurons in the last linear dense layer are the number of unique symptoms with the Sigmoid function. However, the highest five probable symptoms are selected and mapped to the ICD-10 codes. At Altibbi, the tenth Australian modified version of the ICD-10-AM codes is used. Fig. 6 shows the structure of the methodology, where n is the number of labels.

Conventionally, for training the classifier, the fine-tuning dataset was divided into three proportions: 80%, 10%, and 10% for training, validating, and testing, respectively. This division is very common in the literature [33, 53].

Regarding the environmental settings, the Keras [51] deep-learning framework is used. Also, all experiments were implemented using Python (3.7) on Ubuntu-1804-bionic-64 cloud server, the memory is 64 GB, and the processor is Intel(R) Core(TM) i7-7700 with a speed of 3.6 GHz. Whereas, the GPU model is GeForce GTX 1080 of 8 GB.

4.2. Results and discussion

This section discusses the results of BiLSTM at different embedding dimensions and models in terms of precision, recall, and customized hit rate.

Table 5. The evaluation results based on precision and recall.

Model	Dim.	Metric	G1	G2	G3	G4	G5
Word2Vec-SG	50	Precision	0.266	0.310	0.319	**0.351**	0.291
		Recall	0.527	0.453	0.402	0.371	0.291
Word2Vec-SG	100	Precision	0.267	**0.312**	**0.328**	0.331	**0.320**
		Recall	0.541	0.468	**0.410**	**0.384**	**0.326**
Word2Vec-SG	200	Precision	**0.268**	**0.312**	0.321	0.345	0.291
		Recall	**0.544**	**0.472**	0.405	0.381	0.291
Word2Vec-CBOW	50	Precision	0.255	0.304	0.308	0.337	0.309
		Recall	0.516	0.453	0.393	**0.384**	0.309
Word2Vec-CBOW	100	Precision	0.259	0.300	0.309	0.344	0.291
		Recall	0.518	0.450	0.392	0.369	0.286
Word2Vec-CBOW	200	Precision	0.257	0.301	0.304	0.325	0.269
		Recall	0.517	0.448	0.389	0.374	0.263
AraBERT-base-v1	768	Precision	0.243	0.304	0.312	0.330	0.291
		Recall	0.481	0.441	0.395	0.369	0.291
AraBERT-base-v2	768	Precision	0.232	0.290	0.306	0.321	0.291
		Recall	0.444	0.418	0.385	0.364	0.291

Table 6. The performance of the BiLSTM model over groups of data based on the number of symptoms.

Model	Dim.	Group	Hit Rate					
			P1	P2	P3	P4	P5	Overall
Word2Vec-SG	50	G1	0.541	-	-	-	-	
		G2	0.698	0.238	-	-	-	
		G3	0.781	0.378	0.072	-	-	
		G4	0.898	0.489	0.176	0.011	-	
		G5	0.771	0.514	0.171	0.000	0.000	0.532
Word2Vec-SG	100	G1	0.541	-	-	-	-	
		G2	0.696	0.240	-	-	-	
		G3	0.791	0.371	0.066	-	-	
		G4	0.881	0.477	0.153	0.023	-	
		G5	0.771	0.571	0.229	0.029	0.000	0.532
Word2Vec-SG	200	G1	0.544	-	-	-	-	
		G2	0.708	0.236	-	-	-	
		G3	0.779	0.358	0.077	-	-	
		G4	**0.898**	0.443	**0.165**	0.017	-	
		G5	0.771	0.457	0.200	0.029	0.000	0.535
Word2Vec-CBOW	50	G1	0.516	-	-	-	-	
		G2	0.684	0.221	-	-	-	
		G3	0.779	0.343	0.057	-	-	
		G4	0.869	0.483	**0.165**	0.017	-	
		G5	0.771	0.486	0.229	**0.057**	0.000	0.508
Word2Vec-CBOW	100	G1	0.524	-	-	-	-	
		G2	0.686	0.225	-	-	-	
		G3	0.771	0.347	0.056	-	-	
		G4	0.875	0.443	0.148	**0.023**	-	
		G5	0.743	0.486	0.200	0.029	0.000	0.515
Word2Vec-CBOW	200	G1	0.517	-	-	-	-	
		G2	0.683	0.222	-	-	-	
		G3	0.770	0.352	0.055	-	-	
		G4	0.892	0.449	0.153	0.017	-	
		G5	0.771	0.400	0.143	0.029	0.000	0.509
AraBERT-base-v1	768	G1	0.545	-	-	-	-	
		G2	0.719	0.262	-	-	-	
		G3	**0.813**	**0.425**	0.084	-	-	
		G4	0.795	0.384	0.086	0.016	-	
		G5	**0.872**	**0.615**	0.154	0.026	0.000	0.537
AraBERT-base-v2	768	G1	**0.574**	-	-	-	-	
		G2	**0.730**	**0.265**	-	-	-	
		G3	0.806	0.391	**0.088**	-	-	
		G4	0.881	0.483	0.148	0.017	-	
		G5	0.771	0.571	**0.257**	**0.057**	0.000	**0.564**

Table 5 presents the precision and recall at different embedding models and different dimensions across G1, G2, G3, G4, and G5. The best results are highlighted with boldface style. The Word2Vec at the SG structure shows the best precision at dimension (200) at G1 with (26.8%). At G2, the SG structure attained the same results at dimensions: 100 and 200 with a precision of 31.2%. Regarding G3 and G5, the best results obtained when the dimension is 100 with the precision are 32.8%, 32.0%, respectively. The G4 group gained the best precision when the SG structure was 50 with 35.1%. Regarding the recall, G1 and G2 obtained the best results when the SG dimension is 200 with values of 54.4% and 47.2%, respectively. However, for G3, G4, and G5; the best recall is obtained when the dimension is 100 with recall values of 41.0%, 38.4%, and 32.6%. Generally, when comparing the SG with the CBOW structure, the SG demonstrated a better performance in terms of precision and recall. In contrast, in the two versions of the AraBERT model, AraBERT showed a slight decrease in the performance over precision and recall.

Table 6 displays the results of the customized hit rate metric for BiLSTM at AraBERT, different structures and dimensions of Word2Vec, and across groups of data. The acronym $G1$ denotes the group of data with

one actual symptom, $P1$ is the probability of predicting this one symptom correctly, $P2$ is the probability of predicting the two symptoms correctly, and so on. Regarding the overall hit rate, the SG structure of Word2Vec at dimension 200 achieved 53.5%. Whereas AraBERT-base-v1 attained a slightly highest hit rate of 53.7%, and AraBERT-base-v2 attained the highest of 56.4%. Even that, regarding Word2Vec, the SG structure at dimensions 50 and 100 performed slightly the same by having 53.2%. Roughly, the SG structure was better than the CBOW structure. Whilst, AraBERT was better than the SG and CBOW. Considering Group (1) of the dataset, the probability of predicting the actual symptom correctly was the highest at the second version of AraBERT base model (57.4%). The SG at dimension 200 achieved almost similar to AraBERT by having 54.4%. Similarly was the AraBERT-base-v2 which had a hit rate of 54.5%. Also, the SG at dimensions 50 and 100 was gaining relatively close to (54.1%). Increasing the dimension of the SG model did not dramatically influence the hit rate at $P1$. Comparably is for the CBOW, where the best was at dimension 100 of 52.4%.

For the second group (G2), the AraBERT-base-v2 model achieved the best hit rate at $P1$ of 73.0%. Whereas AraBERT-base-v1 attained a hit rate of 71.9%. Besides, the SG structure at dimension 200 gained a recall metric of 70.8%. The SG structure at lower dimensions demonstrated a trivial decrease which was having approximately 69% of recall. Regarding the CBOW structure at G2 and $P1$, it attained slightly less recall metric than the SG by having roughly (68%). Generally at $P2$ regardless of the embedding structure or dimension, predicting the actual two symptoms correctly is considerably less, which is on average 23% for the Word2Vec and 26% for AraBERT. The highest hit rate at $P2$ was 26.5% by AraBERT-base-v2. For the third group (G3), when the number of symptoms increases, the recall decreases. The highest recall at $P1$ was by the AraBERT-base-v1 of 81.3%, while its second version was 80.6%. Next is the SG structure at dimension 100 (79.1%). Generally, when comparing the SG structure with the CBOW, the SG achieved better performance than the CBOW with a recall of an average of (77.3%). For $P2$, the SG achieved almost 36.9% regardless of the dimension, while the CBOW achieved an average of 34.7%. However, the AraBERT-base-v1 gained the best of a hit rate of 42.5%. As it might be anticipated, $P3$ has lower recall values that are less than 10%. Considering what has been mentioned, AraBERT achieved at least better than others. Predicting the three actual symptoms correctly in regards to $G3$ is much harder for the model. However, having one symptom predicted correctly is of a recall of about 81%.

Regarding Group (4), the probability of $P1$ is generally higher than other groups with fewer symptoms. On average, the hit rate was 87.3%. Further, it was the highest at $P1$ by the SG structure of 89.8%. Markedly, dimensions 50 and 200 obtained the same maximum recall (89.8%). The CBOW performed approximately similar to the SG by having 89.2% at dimension 200. Whereas, the AraBERT-base version 1 and 2 achieved 79.5%, 88.1%, respectively. Regarding $P2$, it was better on average than $P2$ for the groups of fewer symptoms. Almost, the average performance of the models at $P2$ is 45.6%. The SG structure at dimension 50 attained the best hit rates at $P2$ and $P3$, which were 48.9%, and 17.6%, respectively. For $P4$, the probabilities decrease dramatically to less than 2%. Finally, at Group (5) and $P1$, the average recall was 78.0%, where the highest was 87.2% by the AraBERT-base-v1 model. At $P2$, almost the recall was 51.3%, while it was the highest by the AraBERT-base-v1 model of 61.5%. For $P3$, the AraBERT-base-v2 achieved the best hit rate of 25.7%. The SG (dimension = 100) and CBOW (dimension = 50) obtained the same and the posterior best hit rate of 22.9%. Regarding $P4$, the recall metric declined to approximately less than 6%. Drastically, at $P5$ the model failed to predict all the five symptoms correctly.

To conclude, for the Word2Vec model, the SG structure showed slightly better performance in terms of hit rate than the CBOW. Furthermore, optimizing the dimension parameter did not improve significantly the hit rate, where it was fluctuating smoothly. Moreover, regardless of the number of actual symptoms, the model was able to predict correctly at least one correct symptom for most of the consultations. In addition to this, the evaluation of AraBERT and Word2Vec shows comparable results for AraBERT at $G1$, $G2$, $G3$, and $G5$ for $P1$ and $P2$ considering the customized hit rate metric.

Table 7. The precision and recall based on the manual evaluation. **Keys:** A.S.C is the average number of symptoms identified correctly per consultation, and A.L.O.S is at least one symptom identified correctly.

Metric	Value
Recall	0.706
Precision	0.233
A.S.C	1.164
A.L.O.S	0.711

The results of the manual evaluation which was conducted by expert doctors are presented in Table 7. It shows the overall precision and recall of 500 evaluated consultations. The model performed successfully with an overall recall of 70.6% and precision of 23.3%. The low precision is due to that it is always computed by dividing by 5 regardless of the number of actual symptoms assigned to the corresponding consultations. Moreover, the model could predict at least one symptom correctly (71.1%) over the entire dataset disregarding the number of actual symptoms. Given that the average number of actual symptoms in the 500 consultations is 1.593, whereas, the average number of identified symptoms by the model is 1.164. This shows a promising ability of the model in predicting and identifying the possible symptoms in such rich and multi-dialectical language.

5. Conclusion

This work targeted the identification of symptoms for text-based medical consultations in a multi-dialect Arabic language context. Altibbi as a digital health platform was referred to as a case study and a source of a huge number of consultations. Two datasets were used, one for learning the embedding, and the other for training the BiLSTM classification model. The AraBERT is used to construct contextual embedding and is compared with the Word2Vec at different dimensions and structures. Evaluating the model based on precision, recall, and a customized hit rate revealed a successful ability to predict at least one of the symptoms correctly with slightly superior performance with the AraBERT embedding. The experiments also show that relying merely on the text of the medical questions as a source of features showed salient results to proceed into actual deployment. As future work, pre-training the BERT model from scratch using Altibbi's content and using it as an embedding model is of significant interest to improve the performance of the symptom extraction model. Also, enhancing the model's performance with other sources of data such as chat and call information is envisaged to boost the model capability.

Declarations

Author contribution statement

Hossam Faris: Conceived and designed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Mohammad Faris: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data.

Maria Habib; Conceived and designed the experiments; Wrote the paper.

Alaa Alomari: Contributed reagents, materials, analysis tools or data.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data availability statement

The data that has been used is confidential.

Declaration of interests statement

The authors declare no conflict of interest.

References

[1] R.F. Mansour, N.O. Aljehane, An optimal segmentation with deep learning based inception network model for intracranial hemorrhage diagnosis, Neural Comput. Appl. (2021) 1–13.

[2] B. Wu, A. Liang, H. Zhang, T. Zhu, Z. Zou, D. Yang, W. Tang, J. Li, J. Su, Application of conventional uav-based high-throughput object detection to the early diagnosis of pine wilt disease by deep learning, For. Ecol. Manag. 486 (2021) 118986.

[3] W.-J. Chang, L.-B. Chen, M.-C. Chen, Y.-C. Chiu, J.-Y. Lin, Scalpeye: a deep learning-based scalp hair inspection and diagnosis system for scalp health, IEEE Access 8 (2020) 134826–134837.

[4] M. Polignano, V. Suriano, P. Lops, M. de Gemmis, G. Semeraro, A study of machine learning models for clinical coding of medical reports at codiesp 2020, in: CLEF (Working Notes), 2020.

[5] A. Chen, L. Zhu, H. Zang, Z. Ding, S. Zhan, Computer-aided diagnosis and decision-making system for medical data analysis: a case study on prostate mr images, J. Manag. Sci. Eng. 4 (4) (2019) 266–278.

[6] B.G. Arndt, J.W. Beasley, M.D. Watkinson, J.L. Temte, W.-J. Tuan, C.A. Sinsky, V.J. Gilchrist, Tethered to the ehr: primary care physician workload assessment using ehr event log data and time-motion observations, Ann. Fam. Med. 15 (5) (2017) 419–426.

[7] H. Faris, M. Habib, M. Faris, H. Elayan, A. Alomari, An intelligent multimodal medical diagnosis system based on patients' medical questions and structured symptoms for telemedicine, Inform. Med. Unlocked 23 (2021) 100513.

[8] C. Sinsky, L. Colligan, L. Li, M. Prgomet, S. Reynolds, L. Goeders, J. Westbrook, M. Tutty, G. Blike, Allocation of physician time in ambulatory practice: a time and motion study in 4 specialties, Ann. Intern. Med. 165 (11) (2016) 753–760.

[9] M. Alshayeji, S. Sultan, et al., Diacritics effect on Arabic speech recognition, Arab. J. Sci. Eng. 44 (11) (2019) 9043–9056.

[10] J. Jiang, H. Zhang, C. Dai, Q. Zhao, H. Feng, Z. Ji, I. Ganchev, Enhancements of attention-based bidirectional lstm for hybrid automatic text summarization, IEEE Access (2021).

[11] H. Faris, M. Habib, M. Faris, A. Alomari, P.A. Castillo, M. Alomari, Classification of Arabic healthcare questions based on word embeddings learned from massive consultations: a deep learning approach, J. Ambient Intell. Humaniz. Comput. (2021) 1–17.

[12] W. Antoun, F. Baly, H. Hajj, Arabert: transformer-based model for Arabic language understanding, arXiv preprint, arXiv:2003.00104.

[13] M. Habib, M. Faris, A. Alomari, H. Faris, Altibbivec: a word embedding model for medical and health applications in the Arabic language, IEEE Access 9 (2021) 133875–133888.

[14] H. Zhang, D. Hu, H. Duan, S. Li, N. Wu, X. Lu, A novel deep learning approach to extract Chinese clinical entities for lung cancer screening and staging, BMC Med. Inform. Decis. Mak. 21 (2) (2021) 1–12.

[15] Y. Mu, H.R. Tizhoosh, R.M. Tayebi, C. Ross, M. Sur, B. Leber, C.J. Campbell, A bert model generates diagnostically relevant semantic embeddings from pathology synopses with active learning, Commun. Med. 1 (1) (2021) 1–13.

[16] K. Sugimoto, T. Takeda, J.-H. Oh, S. Wada, S. Konishi, A. Yamahata, S. Manabe, N. Tomiyama, T. Matsunaga, K. Nakanishi, et al., Extracting clinical terms from radiology reports with deep learning, J. Biomed. Inform. 116 (2021) 103729.

[17] J. Hammoud, A. Vatian, N. Dobrenko, N. Vedernikov, A. Shalyto, N. Gusarova, New Arabic medical dataset for diseases classification, arXiv preprint, arXiv:2106.15236.

[18] H. Faris, M. Habib, M. Faris, M. Alomari, A. Alomari, Medical speciality classification system based on binary particle swarms and ensemble of one vs. rest support vector machines, J. Biomed. Inform. 109 (2020) 103525.

[19] F. Youbi, N. Settouti, Analysis of machine learning and deep learning frameworks for opinion mining on drug reviews, Comput. J. (2021).

[20] M. Habib, M. Faris, R. Qaddoura, A. Alomari, H. Faris, A predictive text system for medical recommendations in telemedicine: a deep learning approach in the Arabic context, IEEE Access (2021).

[21] M. Bi, Q. Zhang, M. Zuo, Z. Xu, Q. Jin, Bi-directional lstm model with symptoms-frequency position attention for question answering system in medical domain, Neural Process. Lett. 51 (2) (2020) 1185–1199.

[22] S.R. Vadyala, E.A. Sherer, Natural language processing accurately categorizes indications, findings and pathology reports from multicenter colonoscopy, arXiv preprint, arXiv:2108.11034.

[23] Z. Xu, S. Lin, J. Chen, Y. Sheng, L. Chen, A semi-supervised method for extracting multiple relations of adverse drug events from biomedical literature, in: 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), vol. 5, IEEE, 2021, pp. 934–938.

[24] L.N. Harfiya, C.-C. Chang, Y.-H. Li, Continuous blood pressure estimation using exclusively photopletysmography by lstm-based signal-to-signal translation, Sensors 21 (9) (2021) 2952.

[25] A. Naglah, F. Khalifa, R. Khaled, A.A.K. Abdel Razek, M. Ghazal, G. Giridharan, A. El-Baz, Novel mri-based cad system for early detection of thyroid cancer using multi-input cnn, Sensors 21 (11) (2021) 3878.

[26] W. Bao, H. Lin, Y. Zhang, J. Wang, S. Zhang, Medical code prediction via capsule networks and icd knowledge, BMC Med. Inform. Decis. Mak. 21 (2) (2021) 1–12.

[27] M. Polignano, M. de Gemmis, G. Semeraro, Comparing transformer-based ner approaches for analysing textual medical diagnoses, in: CLEF eHealth, 2021.

[28] Y. Jalali, M. Fateh, M. Rezvani, V. Abolghasemi, M.H. Anisi, Resbcdu-net: a deep learning framework for lung ct image segmentation, Sensors 21 (1) (2021) 268.

[29] F. Shahidi, Breast cancer histopathology image super-resolution using wide-attention gan with improved Wasserstein gradient penalty and perceptual loss, IEEE Access 9 (2021) 32795–32809.

[30] R.G. Jackson, R. Patel, N. Jayatilleke, A. Kolliakou, M. Ball, G. Gorrell, A. Roberts, R.J. Dobson, R. Stewart, Natural language processing to extract symptoms of severe mental illness from clinical text: the clinical record interactive search comprehensive data extraction (cris-code) project, BMJ Open 7 (1) (2017) e012012.

[31] N. Du, K. Chen, A. Kannan, L. Tran, Y. Chen, I. Shafran, Extracting symptoms and their status from clinical conversations, arXiv preprint, arXiv:1906.02239.

[32] A.S. Eisman, N.R. Shah, C. Eickhoff, G. Zerveas, E.S. Chen, W.-C. Wu, I.N. Sarkar, Extracting angina symptoms from clinical notes using pre-trained transformer architectures, in: AMIA Annual Symposium Proceedings, vol. 2020, American Medical Informatics Association, 2020, p. 412.

[33] R.E. Leiter, E. Santus, Z. Jin, K.C. Lee, M. Yusufov, I. Chien, A. Ramaswamy, E.T. Moseley, Y. Qian, D. Schrag, et al., Deep natural language processing to identify symptom documentation in clinical notes for patients with heart failure undergoing cardiac resynchronization therapy, J. Pain Symp. Manag. 60 (5) (2020) 948–958.

[34] C.-S. Wu, C.-J. Kuo, C.-H. Su, S.-H. Wang, H.-J. Dai, Using text mining to extract depressive symptoms and to validate the diagnosis of major depressive disorder from electronic health records, J. Affect. Disord. 260 (2020) 617–623.

[35] M.Z. Uddin, K.K. Dysthe, A. Følstad, P.B. Brandtzaeg, Deep learning for prediction of depressive symptoms in a large textual dataset, Neural Comput. Appl. (2021) 1–24.

[36] J. Wang, N. Abu-el Rub, J. Gray, H.A. Pham, Y. Zhou, F.J. Manion, M. Liu, X. Song, H. Xu, M. Rouhizadeh, et al., Covid-19 signsym: a fast adaptation of a general clinical nlp tool to identify and normalize Covid-19 signs and symptoms to omop common data model, J. Am. Med. Inform. Assoc. 28 (6) (2021) 1275–1283.

[37] A. Magge, K. O'Connor, M. Scotch, G. Gonzalez-Hernandez, Seed: Symptom extraction from English social media posts using deep learning and transfer learning, medRxiv.

[38] X. Yao, G. Yu, J. Tang, J. Zhang, Extracting depressive symptoms and their associations from an online depression community, Comput. Hum. Behav. 120 (2021) 106734.

[39] D. Guo, M. Li, Y. Yu, Y. Li, G. Duan, F.-X. Wu, J. Wang, Disease inference with symptom extraction and bidirectional recurrent neural network, in: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2018, pp. 864–868.

[40] M. Abulaish, M.A. Parwez, et al., Disease: a biomedical text analytics system for disease symptom extraction and characterization, J. Biomed. Inform. 100 (2019) 103324.

[41] A. Schäfer, N. Blach, O. Rausch, M. Warm, N. Krüger, Towards automated anamnesis summarization: bert-based models for symptom extraction, arXiv preprint, arXiv: 2011.01696.

[42] M. Polignano, F. Narducci, A. Iovine, C. Musto, M. De Gemmis, G. Semeraro, Healthassistantbot: a personal health assistant for the Italian language, IEEE Access 8 (2020) 107479–107497.

[43] S. Wada, R. Iida, K. Torisawa, T. Takeda, S. Manabe, Y. Matsumura, Extracting symptom names and disease-symptom relationships from web texts using a multi-column convolutional neural network, in: MedInfo, 2019, pp. 423–427.

[44] C. Faviez, P. Foulquié, X. Chen, A. Mebarki, S. Quennelle, N. Texier, S. Katsahian, S. Schuck, A. Burgun, Fuzzy matching for symptom detection in tweets: application to Covid-19 during the first wave of the pandemic in France, in: Public Health and Informatics, IOS Press, 2021, pp. 896–900.

[45] N.S. Alghamdi, H.A.H. Mahmoud, A. Abraham, S.A. Alanazi, L. García-Hernández, Predicting depression symptoms in an Arabic psychological forum, IEEE Access 8 (2020) 57317–57334.

[46] S. Alotaibi, R. Mehmood, I. Katib, O. Rana, A. Albeshri, Sehaa: a big data analytics tool for healthcare symptoms and diseases detection using Twitter, apache spark, and machine learning, Appl. Sci. 10 (4) (2020) 1398.

[47] G. Madjarov, D. Kocev, D. Gjorgjevikj, S. Džeroski, An extensive experimental comparison of methods for multi-label learning, Pattern Recognit. 45 (9) (2012) 3084–3104.

[48] P. Yang, X. Sun, W. Li, S. Ma, W. Wu, H. Wang, Sgm: sequence generation model for multi-label classification, arXiv preprint, arXiv:1806.04822.

[49] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: pre-training of deep bidirectional transformers for language understanding, arXiv preprint, arXiv:1810.04805.

[50] T. Mikolov, K. Chen, G. Corrado, J. Dean, L. Sutskever, G. Zweig, word2vec, https://code.google.com/archive/p/word2vec/, 2013, 22.

[51] F. Chollet, et al., Keras, https://keras.io, 2015.

[52] M. Silberztein, F. Atigui, E. Kornyshova, E. Métais, F. Meziane, Natural Language Processing and Information Systems: 23rd International Conference on Applications of Natural Language to Information Systems, NLDB 2018, Paris, France, June 13-15, 2018, Proceedings, Vol. 10859, Springer, 2018.

[53] G. Rao, W. Huang, Z. Feng, Q. Cong, Lstm with sentence representations for document-level sentiment classification, Neurocomputing 308 (2018) 49–57.