A hybrid deep learning approach for phenotype prediction from clinical notes

Sahar Khalafi1 · Nasser Ghadiri1 · Milad Moradi2

Received: 10 December 2021 / Accepted: 9 February 2023 / Published online: 1 March 2023
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract
Identifying patient cohorts from clinical notes in secondary use of electronic health records is a fundamental task in clinical information management. The patient cohort identification process requires identifying the patient phenotypes. However, the growing number of clinical notes makes it difficult to analyze the data manually. Therefore, automatically extracting clinical concepts is crucial to identify patient phenotypes correctly. This paper’s proposed hybrid model is based on a neural bidirectional sequence model (BiLSTM or BiGRU) and a Convolutional Neural Network (CNN) for identifying patients’ phenotypes in clinical notes. Furthermore, an extra CNN layer is run parallel to the hybrid proposed model to extract more features related to each phenotype. We used pre-trained embeddings such as FastText and Word2vec separately as the input layers to evaluate other embedding's performance in identifying patient phenotypes. We also measured the effect of adding additional data cleaning steps on discharge reports to identify patient phenotypes using deep learning models. Results demonstrated the proposed hybrid model extracts more features than existing methods of patient phenotype extraction and provides a better F1-score. We show that complementing the proposed hybrid model with an extra CNN in identifying different phenotypes improves the F1 scores. In addition, eliminating punctuation, numbers, and stop words in discharge reports before training hybrid models increased model performance.

Keywords Clinical notes classification · Deep learning · Natural language processing · Phenotypes prediction · Patient phenotyping · Convolution neural network · Gated recurrent unit

1 Introduction
Electronic Health Records (EHRs) are an integrated set of clinical data related to individual patients. The primary purpose of EHR is to store information for patients’ primary care and healing processes (Nguyen et al. 2019). A crucial task in secondary purposes of EHR is correctly recognizing patient phenotypes to identify the patient cohort (Alzoubi et al. 2019). Identifying patient cohorts in clinical notes is crucial for comparing laboratory tests and treatments within patient cohorts and reducing clinical trials by physicians (Gehrmann et al. 2018). The rapid increase in generating clinical notes in EHR has led to massive clinical information. The lack of efficient methods and tools for analyzing extensive amounts of information has led to complexity, information loss, uncertainty, and prolonged data analysis. Finally, it leads to difficulty in providing appropriate treatment by physicians (Moharasan and Ho 2019).

The Natural Language Processing (NLP) methods based on deep learning models have shown efficiency in analyzing and extracting critical information related to patient phenotypes from clinical notes (Deng et al. 2020). An advantage of using deep learning models is learning features without dictionaries that automatically contain clinical terms (Solas et al. 2020). They are also easily generalizable compared with traditional machine learning models. In a comparative study to extract patient phenotypes from discharge reports...
in MIMIC III\textsuperscript{1}, Gehrmann et al. (2018) show that CNN networks performed better at identifying phenotypes than concept extraction methods. Deep learning methods require no dictionaries and ontologies to learn patients’ phenotypes and increase the model’s accuracy. They executed CNN on each unit phenotype as binary classification. They used pre-trained embeddings as an input layer with word2vec (Mikolov et al. 2013) on all discharge reports in the MIMIC III database. Limiting the space of word search in CNN was also a challenge that was later addressed by Yang et al. (2020). Yang et al. expanded the CNN network by adding input at the sentence level with considering the relationships between words and sentences in extracting patient phenotypes. Marika Cusick et al. (2021) used CNN to identify current suicidal ideation from clinical notes. They showed that the CNN could identify suicidal ideation traits with 94% accuracy and perform better than other machine learning methods, including SVM and the Naive Bayes classifier.

The CNNs and Recurrent Neural Networks (RNNs) are two commonly used deep learning architectures, which have also played an essential role in clinical notes classification (Solares et al. 2020). CNN could exploit n-grams to encode local semantic features. Variations of RNNs such as GRU\textsuperscript{2} (Chung et al. 2014) and LSTM\textsuperscript{3} (Hochreiter and Schmidhuber 1997) capture the semantic structure of long-term dependencies. CNN cannot record term sequences, and sequential models cannot detect keywords in a note. Liu et al. (2019a, b) presented a CNN model as a binary classifier for predicting readmission of heart failure patients from discharge reports. They use word2vec embeddings trained on PubMed articles as the input layer, and the results outperformed random forest or similar machine learning models.

CNN-based models for detecting features just encode local correlations regardless of the sequence of words and grammar, preventing the capture of the semantic structure of sequences of words. On the other hand, sequence models can capture semantic structure in long-term dependencies. Hashir Khan (Khan 2019) used a CNN-LSTM model with FastText (Bojanowski et al. 2017) embeddings as the input layer to predict mortality from discharge reports, providing better results than disease scoring systems. Segura-Bedmar and Raez (Segura-Bedmar and Raez 2019) presented a hybrid CNN-GRU model to identify phenotypes from clinical notes. The hybrid model performed more accurately than non-hybrid models such as CNN and GRU, but adding a fully connected layer did not affect the results. The advantage of using bidirectional sequence models is that they can encode sentences in both directions, enabling them to record semantic and syntactic information (Liu et al. 2019).

In this research, we present a hybrid bidirectional sequence model with CNN to tackle the clinical notes’ challenges and extract various features to identify patient phenotypes. To the author’s knowledge, no existing model for classifying clinical notes considers the combination of bidirectional sequence models with CNN. We can model patient phenotype identification as a binary or multi-label classification problem. We could present every clinical text with a single phenotype associated with a label in binary classification. In contrast, each text will be associated with a set of phenotypes in a multi-label classification. Because of abbreviations, semantic complexities, synonyms, misspellings, and the lack of a logical structure in clinical notes, creating a model for multi-label classification is more complex. The model should be able to learn and identify more variations of features related to different phenotypes.

A key advantage of our model is the ability to automatically extract and learn patient phenotypes by considering the semantic structure in word order and grammar. It extracts the most salient phrases and clinical concepts as multi-label and binary classification. We also use an extra CNN layer to extract more features related to the patient phenotype. We also propose an extra CNN with a hybrid model to extract more features related to different phenotypes and improve the hybrid model. To further enhance the model, an extra CNN identifies local features in parallel, combining adjacent inputs from the embedding layer. The major contributions of this paper are:

- Investigating the effect of data cleansing steps before training the deep learning models on discharge reports.
- Proposing a hybrid bidirectional sequence model with CNN in discharge reports to identify distinct features by considering the semantic structure in long-term dependencies and extracting semantic information from the semantic structure.
- Adding an extra CNN in parallel with a hybrid model to identify patient phenotypes.
- Evaluating the performance of BiGRU layer versus BiLSTM layer to extract related features to patient phenotypes from discharge reports.

In Sect. 2, we elaborate on the characteristics of our proposed model. The dataset for experiments, baseline models, and evaluation metrics are covered in Sect. 3. Section 4 presents experimental parameters and results. Section 5 concludes the paper and points to future research.

\textsuperscript{1} MIMIC III.
\textsuperscript{2} Gated Recurrent Unit.
\textsuperscript{3} Long Short-Term Memory.
2 The proposed model

We propose a bidirectional sequence model and CNN hybrid to identify patient phenotypes in discharge reports. We also improved the model by an extra CNN to extract more features related to each phenotype. We use pre-trained word embeddings such as word2vec or FastText as input layers fed into the bidirectional sequence model. Since patient phenotypes may be clues in any part of the discharge report, capturing long-term dependencies is essential in identifying patient phenotypes. The bidirectional sequence models can capture the syntactic and semantic structure from discharge reports in both forward and backward directions. In identifying the patient phenotype, it is necessary to extract keywords from the semantic structure. Since bidirectional sequence models cannot extract keywords in parallel, each timestep’s hidden state output in the bidirectional sequence model enters the CNN to extract features from the temporal axis. In contrast, the extra CNN with a top-down filter slider in the input layer produces a feature map from adjacent terms to represent a specific concept and extract more medical concepts assigned to each patient phenotype.

2.1 Preprocessing

Unstructured clinical notes contain imperfections, such as spelling errors, abbreviations, lack of grammatical structure, and noise. Before training the model, we need more complex preprocessing steps to convert texts to usable input. The discharge reports in the MIMIC III database must also be converted to the proper form to identify phenotypes before being used to input different models. We used a two-stage pipeline for preprocessing to improve our model’s performance, each stage containing three steps.

For cleaning the data, we changed all uppercase letters to lowercase to reduce noise and simplify the process. We also removed all punctuation marks, including parentheses, semicolons, periods, as well as numbers. In the second step, we deleted stop words based on the assumption that stop words do not help in extracting patient phenotypes features. In the third step, we lemmatized the words. All words are converted to their lemma, assuming that lemmatizing words can help reduce the model’s complexity and improve performance.

2.2 Hybrid bidirectional sequence model with CNN (S‑Conv‑nm)

The architecture of the proposed model is illustrated in Fig. 1. The first layer in our model is the word embedding layer. Word embeddings enter the bidirectional sequence model to save the sequence of information from left-to-right and right-to-left directions. The output from this layer is fed into CNN to learn and locally extract related features to patient phenotype. The feature vectors obtained from the convolution layer enter the global max-pooling layer to reduce the feature vector to a single value and eliminate the noise to get the maximum features globally from the feature vectors. Finally, the global maximum of features is entered the fully connected layer with a sigmoid activation function and one neuron to predict phenotypes individually. In the following subsections, we describe the proposed model.

2.2.1 Embedding layer

Word embedding represents the words by vectors containing numbers that make texts comprehensible to machine learning algorithms. Word embedding methods such as word2vec (Mikolov et al. 2013) by Google and FastText by Facebook (Bojanowski et al. 2017) have been recently used to resolve the limitations of traditional word representation methods. These methods rely on neural network algorithms (Faris et al. 2021). Word embedding methods can capture semantic similarities between words, as well as morphological,
syntactic information, and interrelationships. The word2vec embedding recognizes each word as an independent unit and creates a vector for every word. FastText deals with each word as n-grams of characters and creates a vector for every n-gram such that the sum of the vectors represents one word. This enables the FastText model to perform better when facing unfamiliar words and understand the word structure (Wang et al. 2019).

For evaluating the performance of different embeddings, we pre-trained the embeddings with FastText and word2vec on discharge reports in the MIMIC III database with the skip-gram model as an input layer to the proposed model. We map each word in the discharge reports to its corresponding numeric vector in the word embedding layer. The output of this layer is a two-dimensional matrix. We separately applied the data cleaning steps on discharge reports before training the embeddings.

2.2.2  Bidirectional sequence model

In a sentence, each word plays an important role. Therefore, we use a bidirectional sequence model (BiLSTM, BiGRU) to capture the contextual information and long-term dependencies from the output of embedding layer vectors in both forward and backward directions.

A central element in LSTM is the memory cell, which uses three gates in the LSTM architecture, including forget and input gates, to update the memory content (Hochreiter and Schmidhuber 1997). The forget gate controls old information to determine how much memory information was erased in the previous state. New input information is controlled by the input gate, which contains two layers as follows. The first layer shows which part of the memory cell’s input information is updated by applying a sigmoid activation function. The second layer decides which input elements are preserved to the memory cell (by generating a vector of previous memory cell values with the tangent activation function).

The output of each gate generated by applying the sigmoid activation function is a number between zero and one. A value of zero means clearing information and not sending it to the next state. In contrast, a value of one refers to sending all information in the current state to the next state. The GRU network is a simplified version of the LSTM architecture; by merging LSTM gates, GRU gates have been reduced to update and reset gates (Chung et al. 2014). The update gate determines how much of the current information, previous information, or a combination of the two will be used in state $t$ without overwriting and deleting the information. This feature maintains the long-term dependency of the information in the previous states and affects later states using memory. The reset gate specifies how much information is deleted from previous steps in the current step.

The sequence models (LSTM, GRU) only keep input information they have already seen, preserving past and future data. We use bidirectional sequence models comprising two forward and backward sequence models to obtain sequential text information from both directions. It helps the model better understand the semantic information in the text and ultimately improves the model’s output. The second layer in our proposed model is the bidirectional sequence model. The embedding layer’s output, including word vectors, is separately entered into the bidirectional sequence model to capture semantic dependencies between words in two directions. At time $t$, a word vector enters the bidirectional sequence model. This word depends on the previous word at time $t-1$. The output generation process at each timestep. At each timestep, the output $h_t$ is calculated using the current input $x_t$ and all previous outputs $h_{t-1}$ by the tangent activation function. The output of each timestep is entered CNN to extract semantic features from the long-term dependency between words. Since GRU is a simplified version of LSTM, we expect less execution time and providing a performance close to LSTM.

2.2.3  Conv1d

The convolutional layers comprise a set of filters, also called kernels. We fed the output of each timestep of the bidirectional sequence model to the conv1d layer. After that, a convolution operation is performed on the temporal axis. Figure 2 shows how the features are extracted through applying the convolution operation by the conv1d layer of the temporal axis. The filters move like a sliding window from top to bottom across the temporal axis and perform the
convolutional operation, which is the element-wise product between the local area and the filter. Finally, all the production values are added together, and we obtain the filtered output.

The filters initially have random weights that are updated during backpropagation to achieve acceptable performance. Each feature extracted from each selected filter enters the pooling layer. The pooling layer uses a static activation function while preserving the most salient phrases to reduce the feature space’s size resulting from convolutional operations. The pooling layer reduces the number of parameters and calculations. It has two models: max-pooling and average-pooling, which refer to selecting the top feature and averaging features from feature vectors. Since not all the extracted features are equally important, using the max-pooling model works better than the average-pooling. In this study, we used the one-dimensional global max-pooling layer to select the global features; this layer’s output is used as the input of the next layer (Cipolla and Pentland 1998). From the CNN point of view, feature vectors can be a semantic representation of the input and enter the output layer, a fully connected layer. Output layer uses the sigmoid activation function on its input features; its output comprises a vector containing predicted labels as $\hat{Y} = [\hat{y}_1, \hat{y}_2, \hat{y}_3, \ldots, \hat{y}_i]$ where $Y_i$ refers to $ith$ phenotype labels.

2.3 The enhanced version of the hybrid model (IS-Conv-nm)

The architecture of the enhanced proposed model is illustrated in Fig. 3. We added a CNN, which works in parallel with the hybrid model. It provides more information to the model for extracting phenotypes as a multi-label classifier. The extra CNN works similar to the CNN in our hybrid model as described in Sect. 2.2.3. The difference is that the convolution operation is performed on the output of the embedding layer. The filters’ width in convolution operation is equal to the word embedding size, and its height is adjustable. We used a one-dimensional convolution for extracting semantic features from each filter. The filters play the role...

![Fig. 3](image-url)
of n-grams in extracting salient phrases related to patient phenotypes. The feature vectors obtained from the hybrid model and extra CNN will enter the global max-pooling layer. The output of the global max-pooling layer of the non-hybrid CNN is fed into a fully-connected layer with a ReLU activation function. The output of the hybrid model is concatenated with the output of non-hybrid CNN. Finally, the features enter a fully connected layer with a sigmoid activation function for predicting patient phenotypes.

3 Experimental evaluation

3.1 Dataset

The MIMIC III database is an extensive database containing EHRs of adult patients in the intensive care unit of Beth Israel Deaconess Medical Center between the years 2001 and 2012 (Johnson et al. 2016). The NOTEEVENTS table is a data table in the MIMIC III database that contains open-source clinical notes, including nursing, medical, radiological, and discharge reports for individual patients. We used annotated discharge reports provided by Gehrmann et al. (Gehrmann et al. 2018) from the NOTEEVENTS table to evaluate our proposed method. Discharge reports contain complete information about patient phenotypes (Gehrmann et al. 2018). The total number of annotated discharge reports is 1610 samples. We assign a binary value for 10 different phenotypes to every report. The dataset was segmented so that 20% of the total discharge reports were considered test data, another 20% for validation, and 60% for model training.

3.2 Evaluation metrics

Since our data are unbalanced and most of the data are negatively labeled, the accuracy metric is not sufficient for the analysis of the model. The accuracy metric divides the number of correctly predicted cases by the total number of samples. This is not reliable for unbalanced data. Therefore, we used F1-score to evaluate the output of the binary classification models, based on recall and precision as shown by Eq. (1), Eq. (2) and Eq. (3).

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{1}
\]

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{2}
\]

\[
\text{F1 - score} = \frac{2(\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}} \tag{3}
\]

where \( TP \), and \( FP \) are positive samples that have been correctly and incorrectly classified, respectively. \( TN \) and \( FN \) denote negative samples that have been correctly and incorrectly classified, respectively. We used F1-micro and F1-macro scores for the final evaluation of the multi-label classification models. Precision and recall should include all labels, so that they extend to a labels suitcase standard and are inserted in the harmonic average. Therefore, they are extended to a multi-label criterion and inserted in the harmonic mean.

\[
\text{Micro - avg Recall} = \frac{\sum_{c=1}^{C} TP_c}{\sum_{c=1}^{C} TP_c + FN_c} \tag{4}
\]

\[
\text{Macro - avg Recall} = \frac{\sum_{c=1}^{C} \text{Recall}_c}{C} \tag{5}
\]

\[
\text{Micro - avg precision} = \frac{\sum_{c=1}^{C} TP_c}{\sum_{c=1}^{C} TP_c + FP_c} \tag{6}
\]

\[
\text{Macro - avg precision} = \frac{\sum_{c=1}^{C} \text{Precision}_c}{C} \tag{7}
\]

\[
\text{Micro - avg F1 - score} = \frac{2(\text{Micro - avg Recall} \times \text{Micro - avg Precision})}{\text{Micro - avg Recall} + \text{Micro - avg Precision}} \tag{8}
\]

\[
\text{Macro - avg F1 - score} = \frac{2(\text{Macro - avg Recall} \times \text{Macro - avg Precision})}{\text{Macro - avg Recall} + \text{Macro - avg Precision}} \tag{9}
\]

\( C \) denotes the number of labels (phenotypes) available for classification; Macro-average calculates the metrics independently for each label and then takes the average. Micro-average calculates the aggregation of contributions of all labels to compute the averaged metric. The macro approach calculates the main elements by considering the same weight between the labels. The micro approach considers all units together and performs the calculation without considering the possible differences between the labels.

3.3 Hyperparameters setting

We optimize the hyperparameter values using the 5-fold-cross-validation and report the best values in Table 1. All models were trained using an early stopping based on no change in validation losses after three epochs. The layers used in all models are one-dimensional.

Figure 4 shows that the best filter size for feature extraction on CNN is 2, which plays the role of 2 g an embedding
A hybrid deep learning approach for phenotype prediction from clinical notes

4509

layer. Figure 5 shows the effect of the number of hidden units of CNN and BiLSTM layers in the proposed model on the F1 score. As the number of units increases from 64 to 512, the F1-score improves. It is noteworthy that increasing the number of neurons from 512 does not significantly increase the F1 score but increases the computational cost and, ultimately, the model’s training time. This study considers the number of hidden units for CNN, LSTM, and GRU to be 512. All the experiments presented in this article were performed on a Google Colab platform with GPU-based computations. The models were implemented using Python, TensorFlow, and Keras.

3.4 Baseline models

This section describes the baseline models used to compare the output of multi-label classification.

- CNN: The model proposed to identify patient phenotypes from discharge reports (Gehrmann et al. 2018).
- ws-CNN: A CNN with three different filter sizes with a combination of word and sentence level embeddings proposed by Yang et al. (2020) to identify patient phenotypes from discharge reports.
- Sequence models, including LSTM and GRU that record long-term dependencies unilaterally, followed by a pooling layer.
- Bidirectional sequence models, such as BiLSTM and BiGRU capture semantic structure in both directions, followed by a pooling layer.
- CNN + LSTM: A hybrid model proposed by Khan (2019) where the output of the CNN layer enters the LSTM layer to predict mortality from discharge reports.
- Bidirectional sequence model + CNN (S-Conv-nm): Our basic proposed model that records long-term dependencies in two directions. The bidirectional sequence model is fed to the CNN to extract local features with a pooling layer.
- Bidirectional sequence model + CNN\|CNN (IS-Conv-nm): Our improved proposed model. Bidirectional sequence model followed by a CNN and pooling layer and, in parallel, an extra CNN followed by a pooling layer, dropout, and a fully connected layer. Finally, the outputs of the two CNN are merged.
4 Results and discussion

In the following, we first fine-tune hyperparameters of the proposed model. Finally, we evaluate the proposed model with the baseline models.

4.1 Data cleaning experimental results

We report multi-label classification results to identify phenotypes by applying each data cleaning step before training word2vec embeddings on discharge reports in Fig. 6. The results are reported for running all models ten times and averaging F1 scores. The first step in cleaning data is to remove punctuation and numbers. The second step is to remove stop words. The third step is to lemmatize words, every step following the previous one. As shown in Fig. 6, applying the first and second data cleaning steps improves the F1 score of the models except for BiLSTM. The BiLSTM model considers the sequence of dependencies to record semantic information. Therefore, the more complete the input information, the more accurate the output. Lemmatizing words enhance the CNN’s output by 0.03% because CNN does not consider long-term dependencies and extracts local features. In the following, all evaluations of the proposed model have been performed by applying the first and second data cleaning steps on discharge reports.

4.2 Comparison of baseline deep learning models with the proposed hybrid model

Comparative results are shown in Table 2. Following the comparison of our results, we showed that the proposed model has a significantly better performance than the experiment models, taking advantage of the bidirectional sequence model and CNN. Our experiments showed that using ws-CNN with the combination of sentences and word-level input does not necessarily improve model performance. CNN performed better than the ws-CNN and LSTM models in identifying patient phenotypes from discharge reports.

Because the discharge reports have grammatical complexities, LSTM models that record information sequences in one direction reduce the model’s performance. CNN extracts the essential n-gram information and performs better than LSTM. The BiLSTM model can capture information in two

![Fig. 6 The effect of applying data cleaning steps in discharge reports before training models](image-url)
directions to improve F1-scores compared to the LSTM model. In hybrid models, the results of CNN + LSTM show that complicating deep learning models does not necessarily improve the ultimate performance of the model. The CNN + LSTM hybrid model first extracts semantic information using CNN and then extracts the dependencies between features with LSTM. The CNN + LSTM hybrid model is less efficient in identifying patient phenotypes than the other models evaluated.

We added a CNN to the bidirectional sequence model in the proposed model. The advantages of both models could be exploited by capturing long-term sequences while extracting local features from these sequences. The model resulted in increasing the F1 scores. As shown in Table 2, we found that using the BiGRU layer instead of BiLSTM in the hybrid model could significantly improve the ultimate model’s performance in identifying patient phenotypes as multi-label classification.

### 4.3 Comparing the basic hybrid model with the improved model

Comparing the improved version of the proposed model with the hybrid model, as shown in Table 3, we found that since multi-label classification is more complex than binary classification, it is necessary to extract more features at different levels to identify each of patient phenotype. More features can be extracted by adding an extra CNN that runs in parallel with the hybrid bidirectional sequence model with CNN. The features extracted from each layer provided more information to the ultimate model, which improved the final performance.

We also measure the performance of the proposed model in identifying patient phenotypes as binary classification with word2vec embeddings and report the results in Table 4. The proposed model performed better than compared models in identifying phenotypes as binary classification.

### 4.4 Evaluating the performance of bidirectional sequence models

As shown in Fig. 7, FastText embeddings in models containing the BiGRU layer performed more efficiently than word2vec embeddings. In contrast, in models containing the BiLSTM layer, word2vec embeddings performed better than FastText. word2vec and FastText embeddings had almost similar results due to multi-label classification in the improved hybrid model. Finally, it can be observed that hybrid models with the BiGRU layer provide higher performance than the hybrid models with
the BiLSTM layer in identifying phenotypes as multi-label classification from discharge reports.

5 Conclusion

We identified patient phenotypes in single and multiple forms using NLP and deep learning in this paper. The experimental results showed that the proposed combined and improved model could extract more features to identify phenotypes better than the compared models. Experiments with data cleaning steps on discharge reports have shown that removing punctuation, numbers, stop words, and lemmatizing words improve CNN output. Eliminating the stop words and lemmatizing led to the performance reduction in the bidirectional sequence model. In hybrid models, only removing punctuation, numbers, and stop words improves the model’s ultimate performance. We also showed that word2vec and FastText embeddings in the proposed model (improved version) have almost the same performance. With models containing the BiLSTM layer, word2vec embeddings performed better than FastText embeddings. In the models that include the BiGRU layer, FastText embeddings performed better.

In future works, we will use context-aware embeddings such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al. 2018) as the input layer. BERT can improve deep learning models’ performance. Context-aware embeddings generate a vector for each word by considering how the word appears; word vectors dynamically change with different contexts. Moreover, we could use phenotype-related concepts in clinical terms dictionary input, such as UMLS to combine with unstructured data. Combined phenotype-related concepts with unstructured data will help to increase the performance of extracting phenotypes.

References

Alzoubi H, Alzubi R, Ramzan N, West D, Al-Hadhrami T, Alazab M (2019) A review of automatic phenotyping approaches using electronic health records. Electronics 8(11):1235
Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Linguistics 5:135–146
Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555
Cipolla R, Pentland A (1998) Computer vision for human-machine interaction. Cambridge University Press
Cusick M, Adekkattup P, Campion TR Jr, Sholle ET, Myers A, Banerjee S, Pathak J (2021) Using weak supervision and deep learning to classify clinical notes for identification of current suicidal ideation. J Psychiatric Res 136:95–102
Deng J, Ma Y, Deng-ao L, Zhao J, Liu Y, Zhang H (2020) Classification of breast density categories based on SE-attention neural networks. Comput Methods Programs Biomed. https://doi.org/10.1016/j.cmpb.2020.105489
Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
Faris H, Habib M, Faris M, Alomari A, Castillo PA, Alomari M (2021) Classification of arabic healthcare questions based on word embeddings learned from massive consultations: a deep learning approach. Journal of Ambient Intelligence and Humanized Computing, 1–17
Gehrmann S, Dernoncourt F, Li Y, Carlson ET, Wu JT, Welt J, Tyler PD (2018) Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives. PLoS ONE 13(2):e0192360
Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
Johnson AE, Pollard TJ, Shen L, Li-Wei HL, Feng M, Ghassemi M, Mark RG (2016) MIMIC-III, a freely accessible critical care database. Sci data 3(1):1–9
Khan MH (2019) A CNN-LSTM for predicting mortality in the ICU. Master’s thesis, University of Tennessee. https://trace.tennessee.edu/utk_graddthes/5441
Liu J, Yang Y, Lv S, Wang J, Chen H (2019a) Attention-based BiGRU-CNN for chinese question classification. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-019-01344-9

Fig. 7 Comparison multi-label classification to identify different phenotypes with FastText and word2vec embeddings
Liu X, Chen Y, Bae J, Li H, Johnston J, Sanger T (2019b) Predicting Heart Failure Readmission from Clinical Notes Using Deep Learning Paper presented at the 2019b IEEE International Conference on Bioinformatics and Biomedicine (BIBM)

Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality Paper presented at the Advances in neural information processing systems

Moharasan G, Ho T-B (2019) Extraction of temporal information from clinical narratives. J Healthcare Inf Res 3(2):220–244

Nguyen BP, Pham HN, Tran H, Nghiem N, Nguyen QH, Do TT, Simpson CR (2019) Predicting the onset of type 2 diabetes using wide and deep learning with electronic health records. Comput Methods Programs Biomed 182:105055

Segura-Bedmar I, Raez P (2019) Cohort selection for clinical trials using deep learning models. J Am Med Inform Assoc 26(11):1181–1188

Solares JRA, Raimondi FED, Zhu Y, Rahimian F, Canoy D, Tran J, Nazarzadeh M (2020) Deep learning for electronic health records: a comparative review of multiple deep neural architectures. J Biomed Inform 101:103337

Wang B, Wang A, Chen F, Wang Y, Kuo C-CJ (2019) Evaluating word embedding models: methods and experimental results. APSIPA Trans Signal Inform Process. https://doi.org/10.1017/ATSIP.2019.12

Yang Z, Dehmer M, Yli-Harja O, Emmert-Streib F (2020) Combining deep learning with token selection for patient phenotyping from electronic health records. Sci Rep 10(1):1–18

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.